Universidade Estadual de Campinas
Instituto de Computação

Ícaro Cavalcante Dourado

Graph-based rank aggregation

Agregação de ranks baseada em grafos

CAMPINAS
2020

Ícaro Cavalcante Dourado

Graph-based rank aggregation

Agregação de ranks baseada em grafos

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Ricardo da Silva Torres

Este exemplar corresponde à versão final da Tese defendida por Ícaro Cavalcante Dourado e orientada pelo Prof. Dr. Ricardo da Silva Torres.

CAMPINAS
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

D748g

Dourado, Icaro Cavalcante, 1985-
Graph-based rank aggregation / Icaro Cavalcante Dourado. – Campinas, SP : [s.n.], 2020.

Orientador: Ricardo da Silva Torres.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas de recuperação da informação. 2. Reconhecimento de padrões. 3. Representações dos grafos. I. Torres, Ricardo da Silva, 1977-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Agregação de ranks baseada em grafos
Palavras-chave em inglês:
Information storage and retrieval systems
Pattern recognition
Representations of graphs
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Ricardo da Silva Torres [Orientador]
Edleno Silva de Moura
Renata de Matos Galante
Júlio César dos Reis
André Santanchè
Data de defesa: 13-03-2020
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-7185-0411
- Currículo Lattes do autor: http://lattes.cnpq.br/1896476549860669

Universidade Estadual de Campinas
Instituto de Computação

Ícaro Cavalcante Dourado

Graph-based rank aggregation

Agregação de ranks baseada em grafos

Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres
  Instituto de Computação - UNICAMP
- Edleno Silva de Moura
  Instituto de Computação - UFAM
- Renata de Matos Galantes
  Instituto de Informática - UFRGS
- Júlio Cesar dos Reis
  Instituto de Computação - UNICAMP
- André Santanchè
  Instituto de Computação - UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 13 de março de 2020

Acknowledgements

First, I thank God for giving me health and strength to achieve this goal.

I am very thankful to my advisor Dr. Ricardo Torres for his support and interest during this research. I also thank him for his valuable and kind words of encouragement and advice, sharing an invaluable academic and personal experience.

I thank my family for their dedication and all the support, specially Maria Aparecida, Daniel, Ana Flávia, and Davi. I also thank them for standing by my side and for the companionship and patience.

I thank the professors and colleagues that contributed to the work, specially Dr. Daniel Pedronette from São Paulo State University, and Dr. Salvatore Tabbone from Université de Lorraine. I also thank my colleagues from RECOD lab who helped with discussion, insights, and support.

I thank Beegol and UNICAMP, where I have been working professionally during and beyond the development of this research, for their support. I also thank the Norwegian University of Science and Technology (NTNU) for its support.

This work was partially funded by CNPq, São Paulo Research Foundation – FAPESP (grants #2014/12236-1, #2015/24494-8, #2016/50250-1, and #2017/20945-0) and the FAPESP-Microsoft Virtual Institute (grants #2013/50155-0 and #2014/50715-9). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Resumo

Neste trabalho, apresentamos uma abordagem robusta de agregação de listas baseada em grafos, capaz de combinar resultados de modelos de recuperação isolados. O método segue um esquema não supervisionado, que é independente de como as listas isoladas são geradas. Nossa abordagem é capaz de incorporar modelos heterogêneos, de diferentes critérios de recuperação, tal como baseados em conteúdo textual, de imagem ou híbridos. Reformulamos o problema de recuperação ad-hoc como uma recuperação baseada em *fusion graphs*, que propomos como um novo modelo de representação unificada capaz de mesclar várias listas e expressar automaticamente inter-relações de resultados de recuperação. Assim, mostramos que o sistema de recuperação se beneficia do aprendizado da estrutura intrínseca das coleções, levando a melhores resultados de busca. Nossa formulação de agregação baseada em grafos, diferentemente das abordagens existentes, permite encapsular informação contextual oriunda de múltiplas listas, que podem ser usadas diretamente para ranqueamento. Experimentos realizados demonstram que o método apresenta alto desempenho, produzindo melhores eficácias que métodos recentes da literatura e promovendo ganhos expressivos sobre os métodos de recuperação fundidos.

Outra contribuição é a extensão da proposta de grafo de fusão visando consulta eficiente. Trabalhos anteriores são promissores quanto à eficácia, mas geralmente ignoram questões de eficiência. Propomos uma função inovadora de agregação de consulta, não supervisionada, intrinsecamente multimodal almejando recuperação eficiente e eficaz. Introduzimos os conceitos de projeção e indexação de modelos de representação de agregação de consulta com base em grafos, e a sua aplicação em tarefas de busca. Formulações de projeção são propostas para representações de consulta baseadas em grafos. Introduzimos os *fusion vectors*, uma representação de fusão tardia de objetos com base em listas, a partir da qual é definido um modelo de recuperação baseado intrinsecamente em agregação. A seguir, apresentamos uma abordagem para consulta rápida baseada nos vetores de fusão, promovendo agregação de consultas eficiente. O método apresentou alta eficácia quanto ao estado da arte, além de trazer uma perspectiva de eficiência pouco abordada. Ganhos consistentes de eficiência são alcançadas em relação aos trabalhos recentes.

Também propomos modelos de representação baseados em consulta para problemas gerais de predição. Os conceitos de grafos de fusão e vetores de fusão são estendidos para cenários de predição, nos quais podem ser usados para construir um modelo de estimador para determinar se um objeto de avaliação (ainda que multimodal) se refere a uma classe ou não. Experimentos em tarefas de classificação multimodal, tal como detecção de inundação, mostraram que a solução é altamente eficaz para diferentes cenários de predição que envolvam dados textuais, visuais e multimodais, produzindo resultados melhores que vários métodos recentes.

Por fim, investigamos a adoção de abordagens de aprendizagem para ajudar a otimizar a criação de modelos de representação baseados em consultas, a fim de maximizar seus aspectos de capacidade discriminativa e eficiência em tarefas de predição e de busca.

Abstract

In this work, we introduce a robust graph-based rank aggregation approach, capable of combining results of isolated ranker models in retrieval tasks. The method follows an unsupervised scheme, which is independent of how the isolated ranks are formulated. Our approach is able to incorporate heterogeneous models, defined in terms of different ranking criteria, such as those based on textual, image, or hybrid content representations. We reformulate the ad-hoc retrieval problem as a graph-based retrieval based on *fusion graphs*, which we propose as a new unified representation model capable of merging multiple ranks and expressing inter-relationships of retrieval results automatically. By doing so, we show that the retrieval system can benefit from learning the manifold structure of datasets, thus leading to more effective results. Our graph-based aggregation formulation, unlike existing approaches, allows for encapsulating contextual information encoded from multiple ranks, which can be directly used for ranking. Performed experiments demonstrate that our method reaches top performance, yielding better effectiveness scores than state-of-the-art baseline methods and promoting large gains over the rankers being fused.

Another contribution refers to the extension of the fusion graph solution for efficient rank aggregation. Although previous works are promising with respect to effectiveness, they usually overlook efficiency aspects. We propose an innovative rank aggregation function that it is unsupervised, intrinsically multimodal, and targeted for fast retrieval and top effectiveness performance. We introduce the concepts of embedding and indexing graph-based rank-aggregation representation models, and their application for search tasks. Embedding formulations are also proposed for graph-based rank representations. We introduce the concept of *fusion vectors*, a late-fusion representation of objects based on ranks, from which an intrinsically rank-aggregation retrieval model is defined. Next, we present an approach for fast retrieval based on fusion vectors, thus promoting an efficient rank aggregation system. Our method presents top effectiveness performance among state-of-the-art related work, while promoting an efficiency perspective not yet covered. Consistent speedups are achieved against the recent baselines in all datasets considered.

Derived from the fusion graphs and fusion vectors, we propose rank-based representation models for general prediction problems. The concepts of fusion graphs and fusion vectors are extended to prediction scenarios, where they can be used to build an estimator model to determine whether an input (even multimodal) object refers to a class or not. Performed experiments in the context of multimodal classification tasks, such as flood detection, show that the proposed solution is highly effective for different detection scenarios involving textual, visual, and multimodal features, yielding better detection results than several state-of-the-art methods.

Finally, we investigate the adoption of learning approaches to help optimize the creation of rank-based representation models, in order to maximize their discriminative power and efficiency aspects in prediction and search tasks.

List of Figures

List of Tables

List of Acronyms

IDCG Ideal Discounted Cumulative Gain 46

Contents

Chapter 1

Introduction

1.1 Motivation

The internet has been increasingly present and necessary in people's lives, either as a source of information, or as an instrument of communication and interaction. The proliferation of digital media and social networks is expanding substantially the volume and diversity of digital content. Huge volumes of complex data, comprising multiple kinds of modalities, have been created continuously (Figure 1.1). In general, these data are heterogeneous, unstructured, unlabeled, and derived from multiple modalities. This scenario presents real challenges in terms of storing, indexing, correlating, analyzing, and retrieving such content.

Such digital content is of great relevance to support the development of retrieval and prediction models. In particular, multimodal data analysis is required in several scenarios, such as Content-Based Information Retrieval (CBIR) [126, 153], and Multimedia Event Detection (MED) of natural disasters (Figure 1.2). Due to the numerous challenges and business opportunities, data analysis involving multimedia and heterogeneous content is a hot topic that attracts a lot of attention not only from public and private sectors, but also from academia. This increased the demand for new approaches in two research venues: (1) effective and efficient search computational methods, and (2) the creation of sophisticated feature extraction algorithms.

Effective and efficient computational methods, such as for retrieval, should be employed to address existing users' information needs. One common solution relies on ad-hoc retrieval (also called content-based retrieval), which allows documents, images, or multimodal objects to be adopted as queries in a search system. Ad-hoc retrieval has been exploited in several applications, such as service providers, digital libraries, and social media.

On the other side, feature extraction algorithms are important as they are the basis of subsequent generalization and learning models, commonly used in several domains, such as search and classification tasks. Proposals of description approaches for images, texts, and multimedia data have advanced in the last decades, leading to more discriminative and effective models for content-based data analysis.

Despite the continuous advance on feature extractors and machine learning techniques, a single descriptor or a single modality is often insufficient to achieve effective prediction

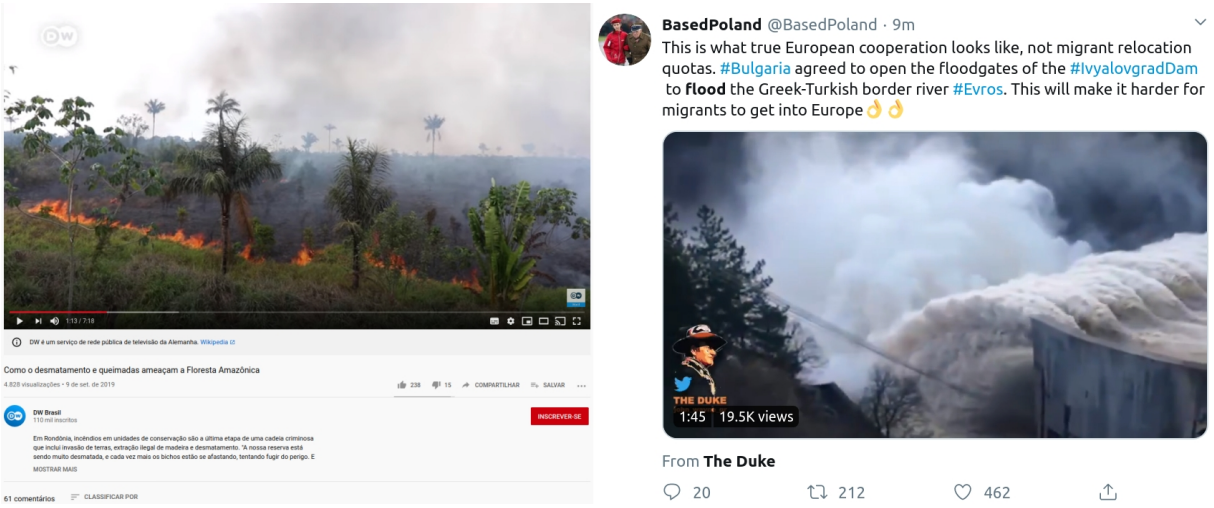
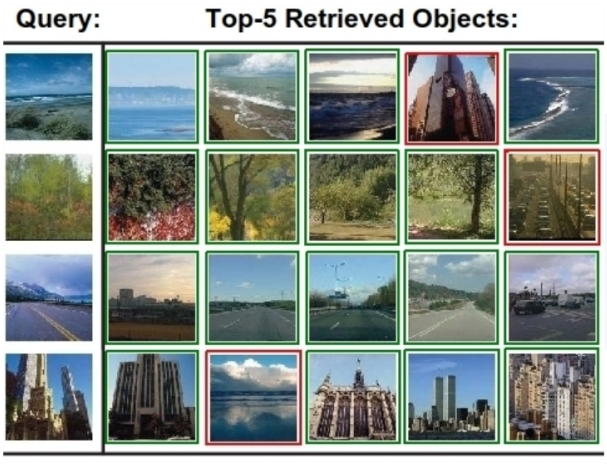


Figure 1.1: Examples of complex digital objects, which can be composed by images, videos, thumbnails, title, description, hyperlinks, tags, timestamp, etc. Pictures taken from YouTube© and Twitter© sites (as of March 12th 2020).



(a) Content-based retrieval.

(b) Multimodal classification.

Figure 1.2: Examples of multimodal data analysis. Pictures in (a) belong to Places dataset [152], whereas picture in (b) was taken from Twitter© (as of March 12th 2020).

results in real case scenarios. Descriptors have specific pros and cons, because each one often focuses on a specific point of view of a single modality. For example, dedicated descriptors may be created to characterize scenes, textual descriptions, movement, symbols, signals, etc. The choice of the most suitable technique often depends on the circumstances (e.g., application or dataset) in which they are used. For this reason, descriptors and retrieval models often provide complementary views, when adopted in combination. In fact, an active research venue relies on exploiting their complementary view, by aggregation, aiming to improve the effectiveness of complex services, such as search, classification, or recommendation.

Scenarios involving heterogeneous data impose a challenge of selecting features or models to combine, which is performed by either supervised or unsupervised approaches. Unsupervised approaches are necessary in the absence of labeled data, which is prominent nowadays, or in scenarios involving lower computation capabilities or large amount of data.

Existing aggregation methods are often categorized as early fusion or late fusion approaches. *Early-fusion* methods emphasize the generation of composite descriptions for samples, whereas *late-fusion* methods perform a combination of techniques focused on a target problem. Majority voting of classifiers and rank aggregation functions are examples of late-fusion methods. Late-fusion methods are especially useful when the raw data from the objects are not available, and are potentially more effective than early-fusion methods because they are specifically designed or optimized for the problem being solved.

Rank aggregation functions allow retrieval models (or rankers) to be built on top of others. They combine results from different rankers and aim to promote more effective retrieval results, without dealing with raw data or low-level descriptors. Besides, even heterogeneous models such as text-based or image-based can be gathered together. Rank aggregation techniques are important in many applications, such as meta-search, document filtering, recommendation systems, social choice, etc. The challenge involving rank aggregation is that obtaining the optimal aggregation is NP-hard for more than three input ranks [48].

Rank aggregation approaches can be categorized according to the use (supervised) or not (unsupervised) of learning methods. Both unsupervised and supervised rank aggregation methods have been proposed. Although supervised methods have the potential to produce better fusions, in practice they demand more computational cost and require training data that may be either unavailable or expensive to obtain. A crowd paradigm, aimed at obtaining labeled training data through voluntary or paid collaborative work, can mitigate the lack of training data. However, this labeling task can still be a time-consuming, expensive, and unfeasible process; or even introduce bias to the data.

Many rank aggregation functions have been proposed under varied approaches, such as supervised learning [96, 97], rank position averaging [17, 32], retrieval score combination [52, 104], Markov Chains [48, 114], and graph of correlations [107, 147]. Among these works, graph-based approaches were the most prominent. As graphs are a flexible and powerful tool for modeling arbitrary structures and relationships among data objects and ranks, the investigation of graph-based approaches for aggregation is one of our main objectives. We claim that ranks and their relationship can be encoded within graphs, and then serve as an underlying structure for object representation.

Although some of these aggregation functions are promising with respect to effectiveness, and some of them also handle multimodality, existing proposals in the field are not usually concerned with efficiency. Nevertheless, information retrieval typically has to deal with large datasets, thus demanding efficient retrieval. On the other hand, a number of works from related research fields have been proposed regarding indexing structures, embedding formulations [23, 44, 116], and solution for approximate search [89]. We want to investigate the applicability of such initiatives in the context of rank aggregation.

Previous graph-based rank aggregation functions usually build a certain graph for modeling the whole collection and solve an optimization function over it [48, 105–107, 114, 147]. They have not focused so far on strategies of graph-based representation models based on ranks. Such approach would open an interesting possibility in which retrieval models could be defined upon those models, as well as they could act as higher-level representation structures for any task.

In fact, several open problems in different applications can be mapped to rank-based solutions, which potentially benefit from complementary views provided by multiple ranking criteria. One research venue comprises the use of rank aggregation solutions for prediction problems related to event recognition. Many research works have been proposed in the literature combining heterogeneous data sources (remotely sensed information and social media) to analyze natural disasters. In [153], authors point out the benefit of exploring and combining multi-modal features with different models. Moreover, combining different kinds of features (local vs holistic) improve substantially the retrieval precision [147], and most retrieval fusion approaches are based on rank fusion [36, 147]. From this perspective, we intend to explore the concept of rank-based representation models in prediction tasks, also contrasted to both early-fusion and late-fusion techniques.

1.2 Hypothesis and Research Questions

In light of the provided context and research perspectives, the main hypothesis addressed in this work is:

> ***Modeling objects using a graph-based representation, say a* fusion graph*, created based on information encoded on multiple ranks, leads to effective and efficient search and prediction systems.***

From this hypothesis, we derive and investigate throughout this work the following Research Questions (RQ's):

RQ$_1$ In a search scenario composed of multiple heterogeneous retrieval models at disposal, is it possible to define an unsupervised representation model, by means of ranks, to represent a query object as a graph, say *fusion graph*, that encodes its ranks and rank relationships effectively?

RQ$_2$ Is fusion graph an appropriate underlying structure to define a competitive unsupervised rank aggregation function when compared to state-of-the-art initiatives?

RQ$_3$ How to make graph-based rank aggregation functions efficient for search scenarios that require fast sub-linear retrieval times?

RQ$_4$ Are graph-based rank representation models feasible for multimodal prediction tasks?

RQ$_5$ When labeled data is available, how could graph-based rank representation models be learned by training, by taking into account their discriminative power and efficiency?

Our intended methodology seeks to answer each Research Question individually, both theoretically and experimentally. We also intend to empathize the contributions of our findings and point out future research directions.

1.3 Contributions

This work provides contributions in different areas, such as content-based information retrieval, rank aggregation, multimodal representation, rank-based fusion and embedding, multimodal retrieval, and multimodal classification.

In order to provide a comprehensive solution to deal with multiple heterogeneous data, descriptors and retrieval models, we define rank-based representation components, such as fusion graphs and fusion vectors, for general applicability. We introduce the notion of rank-based representation as a novel research venue, which is capable of exploring dataset information regardless the presence of labeled information. These two proposals are initially employed in rank aggregation functions, in order to build better content-based retrieval models. The fusion graph is capable of combining results and express inter-relationships of retrieval results automatically, in an unsupervised scheme, which is independent of how the isolated ranks are formulated.

Another contribution is the reformulation of the ad-hoc retrieval problem as a graph-based retrieval based on fusion graphs. This brings a new perspective for ad-hoc retrieval, not yet explored by previous related works. Besides, the retrieval system can benefit from learning the manifold structure of datasets, thus leading to effective results.

We introduce the concepts of embedding and indexing graph-based rank-aggregation representation models. To the best of our knowledge, we are the first to introduce these concepts to this domain. This contributes for fast retrieval in rank aggregation functions, and also establishes a novel unsupervised rank-based representation model.

We validate the application of fusion vectors for search tasks, at first. Next, we advance its application for multimodal prediction scenarios, presenting notable results when compared to the state of the art in early and late fusion methods, either unsupervised and supervised ones. We claim this is a new paradigm for multimodal prediction, to be further advanced in the literature.

Finally, we propose learning approaches to help optimize the creation of rank-based representation models, in order to maximize their discriminative power in prediction and search tasks involving labeled data. We introduce new scoring functions for graphs and present their applicability in rank-based representation models.

1.4 Text Organization

Figure 1.3 outlines the research approach of this work by indicating the chapters, their relationships, and their addressed content and RQ's.

In Chapter 2, we formally provide preliminary definitions based on which we further develop our research, and then we cover and discuss previous works that relate to our research directions.

RQ_1 is a preliminary yet essential question, that also motivates the other addressed research questions. An unsupervised rank-based representation model for general applicability is the main objective. As RQ_2 consists of an experimental validation of RQ_1, we address these two questions together, in Chapter 3. We introduce a graph-based aggregation, then we reformulate the ad-hoc retrieval problem as a graph-based retrieval, and evaluate these proposals experimentally.

Next, RQ_3 motivates us to advance the findings of RQ_1 even further, aiming at efficient retrieval. RQ_3 drives the work presented in Chapter 4. The concept of a fusion vector is presented and explored for search tasks, involving the embedding of fusion graphs into a vector space and their indexing for fast retrieval.

Motivated by the investigation of the applicability of rank aggregation functions and multimodality in multimodal prediction tasks, we address RQ_4 in Chapter 5. We design and validate a methodology to apply fusion graphs and fusion vectors as general-purpose unsupervised representation models for tasks involving multimodal prediction.

Finally, we address RQ_5 in Chapter 6, which aims to define an alternative optimized formulation for fusion vectors (from Chapter 4) based on learning approaches. We discuss a few learning approaches on how to obtain representation models from ranks, then we propose and validate a feature engineering approach in order to optimize the embedding process of fusion graphs.

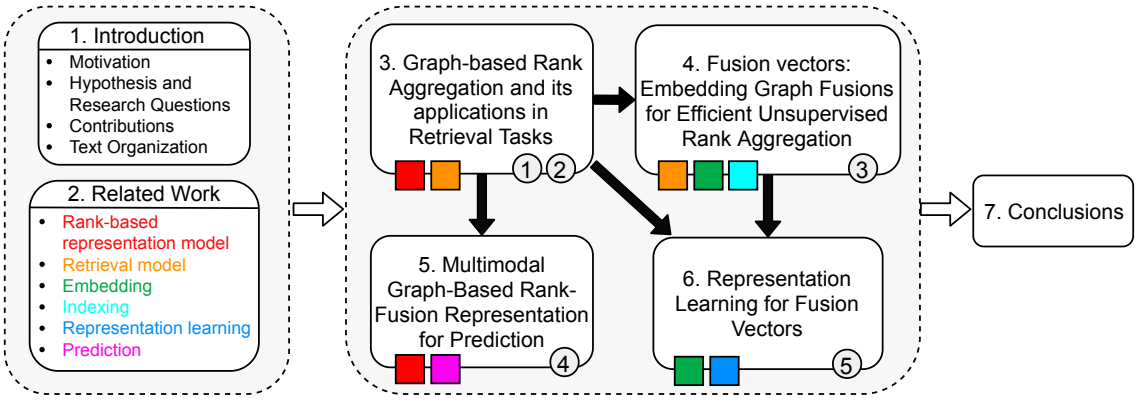


Figure 1.3: Research outline indicating the Research Questions, the chapters and their relationships. The colored squares below each of the main chapters express the presence of the respective topics introduced by Chapter 2. The numbered circles, in turn, indicate which RQ's each chapter addresses.

Chapter 2

Related Work

This chapter introduces background concepts and presents some of the most representative methods proposed in the literature related to our research.

2.1 Preliminary Definitions

Here we formally establish and standardize preliminary definitions based on which we discuss the literature and design our methods.

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a collection of n *digital objects* (or *samples*), where n is the collection size. A sample can be a document, an image, a video, or even a hybrid (called multimodal) object.

Each sample $s \in S$ can be characterized (or described) by a *descriptor* $\mathcal{D} : s \to \epsilon$, which assigns (extracts from the object) to s a vector, matrix, graph, or any other data structure ϵ. The main purpose of such descriptions is to allow the comparison of objects, supporting the creation of services, such as search, recommendation, or prediction. Each descriptor has its own assumption, pros and cons, and represents a specific point of view with respect to the raw data. One descriptor may be particularly specialized for either object detection, scene detection, corner detection, keywords, etc. For this reason, it is common to use multiple descriptors to characterize a collection, due to their complementary view.

A *comparator* $\mathcal{C} : (\epsilon_i, \epsilon_j) \mapsto \varsigma \in \mathbb{R}^+$ is adopted to compare two samples (s_i, s_j) in terms of their descriptions, thus producing a resultant *score* ς. Both underlying similarity or dissimilarity functions can be used, such as the cosine similarity and the Euclidean distance. For similarity-based comparators, higher scores mean more equivalence, and vice-versa. In general, one can perform a standardization procedure to convert dissimilarity into similarity scores, so that heterogeneous comparators can be used together when applied in a broader context.

Let us assume that S is also a *response set* composed of *response items* (retrieved samples), which are retrieved by an information retrieval system. A *search* brings a *rank* τ in response of a *query* q. In ad-hoc retrieval, q follows the same definition of a sample, but refers to the input object in the context of a search. A rank is a permutation of $S_L \subseteq S$, where $L \ll n$ in general, such that τ_q provides the most similar – or equivalently the least dissimilar – response samples from S, to q, in order, according to a relevance

criteria. L is used as a cut-off parameter.

In prediction tasks, a query is referred to *test sample*, and the response set is referred to as *train set*. These terms can be used interchangeably, but a train set is demanded to contain labeled samples while a response set is not.

A *ranker* $R : q \mapsto \tau$ establishes a ranking model and it is composed of a tuple $(\mathcal{D}, \mathcal{C})$. It computes a rank τ for q, regarding S. A ranker may be seen as a simplified retrieval model [9]. For simplicity, in the context a ranker we write $\varsigma(s_i, s_j)$ meaning $\mathcal{C}(\mathcal{D}(s_i), \mathcal{D}(s_i))$ for its $\mathcal{D}$ and $\mathcal{C}$. Let $\rho_{\tau_q}(x)$ be the position of x in τ_q, starting by 1, or just $\rho(x)$ if the rank is clear enough in the context. For a ranker composed of a similarity-based comparator, $\rho(s_i) \leq \rho(s_i)$ if $\varsigma(q, s_i) \leq \varsigma(q, s_j)$. The notation $\varsigma_{\tau_q}(s_i, s_j)$ stands for the general case, which is the score between s_i and s_j with respect to the same descriptor and comparator from the particular ranker that produced the rank τ_q for the query q.

While a ranker establishes a ranking model, different descriptors and comparators can be used to compose rankers, and it is well known that descriptors can be complementary, as well as comparators. Given a set of m rankers, $\{R_1, R_2, \ldots, R_m\}$, being used for retrieval over S, we can obtain for q a rank set $\mathcal{T}_q = \{\tau_1, \tau_2, \ldots, \tau_m\}$, from which a *rank aggregation function* $f : \mathcal{T}_q \mapsto \tau_{q,f}$ produces a combined rank, expected to be more effective than the individual ranks from $\mathcal{T}_q$. $\tau_{q,f}$ denotes the rank produced for q by f in terms of $\mathcal{T}_q$.

We define a *multi-ranked object* as an object represented by its multiple ranks. Different from early-fusion techniques, the object is not represented by features from multiple feature extractors, but from ranks as if the object was a query over multiple rankers. We claim that this approach has benefits, to be discussed throughout this work.

A *graph embedding function* $\mathcal{E}$ defines a d-dimensional vector space and projects graphs on it. A graph $\mathcal{G}(V, E)$, where V is the vertex set and E is the edge set, is projected to that space as a vector $\mathcal{V}$, so that $\mathcal{E} : \mathcal{G} \mapsto \mathcal{V} \in \mathbb{R}^{\mathrm{d}}$, where $d \ll |V|$ ideally. $\mathcal{E}$ is expected to preserve some graph properties and also proximity measures in the resultant space. The d-sized feature set of the vector space is called its *vocabulary* or *codebook*.

An *exact search* – for the query q, dataset S, and ranker $R(\mathcal{D}, \mathcal{C})$ – is asymptotically linear to $|S|$, and can be expressed by:

$$\underset{s_i \in S}{\operatorname{argsort}} \left(\mathcal{C} \left(\mathcal{D} \left(q \right), \mathcal{D} \left(s_i \right) \right) \right), \tag{2.1}$$

where *argsort* is defined as a function that returns the indices that would sort an array. In practice, only a top-L of the response items most similar to q are of interest. *Approximate search*, conversely, works in sublinear time to $|S|$. However, it has a trade-off between recall and complexity: it must be faster than the exact search, while retaining quality as much as possible. It usually adopts *indexing structures* to reduce the search space [89]. The main idea is that some loss is acceptable to make searches faster, specially for large datasets.

An *event predictor*, or only *predictor*, can be modeled as a regression model, $Y \approx g(X, \beta)$, where g is an approximation function, X is a vector representing one or more independent variables, Y is the dependent variable (target), and β represents unknown parameters. A learning model explores the training corpus S to find a g that minimizes an error metric (measured by a *loss function*). If the training samples are labeled, Y

Table 2.1: Notations of our preliminary definitions.

Notation	Meaning
S	collection, or the response set in the context of search, or training set for prediction
n	$\|S\|$
s	a sample, from S
$\mathcal{D}$	descriptor
$\epsilon(s)$	a data structure, generated by $\mathcal{D}$, that describes s
$\mathcal{C}$	comparator
ς	score, of codomain $\mathbb{R}^+$, generated by $\mathcal{C}$ over $(\epsilon(s_i), \epsilon(s_j))$
q	query
L	cut-off parameter
R	ranker, a tuple $(\mathcal{D}, \mathcal{C})$
τ	rank, a permutation of $S_L \subseteq S$, generated by R
τ_q	rank for q
$\varsigma_{\tau_q}(s_i, s_j)$	score between s_i and s_j with respect to the same R that produced τ_q for q
$\rho_{\tau_q}(x)$	position of x in τ_q
m	number of rankers used
$\mathcal{T}_q$	rank set for q, $\{\tau_1, \tau_2, \ldots, \tau_m\}$, generated by $\{R_1, R_2, \ldots, R_m\}$
f	a rank aggregation function
$\tau_{q,f}$	the output rank of f, expressed by $f(\mathcal{T}_q)$
d	the dimensionality of a vector space
$\mathcal{G}(V, E)$	a graph, where V is the vertex set and E is the edge set
$\mathcal{V}$	a vector $\in \mathbb{R}^d$
$\mathcal{E}$	a graph embedding function $\mathcal{E} : \mathcal{G} \mapsto \mathcal{V}$
$E(Y\|X) = g(X, \beta) \approx Y$	event estimator

may be categorical. Still, a regressor $E(Y|X) = g(X, \beta)$ can be built, so that posterior probabilities are adopted as a confidence estimation of a sample to belong to an event type (or class). X can be any variable set that describes the samples.

We summarize the main notations and their definitions in Table 2.1.

2.2 Rank Aggregation

The Kemeny ranking problem is defined as the task of obtaining a consensus rank (or median rank) that best represents a given set of ranks, i.e., an optimal permutation that best summarizes them. Its general case, known as **Rank Aggregation Problem (RAP)**, targets any kind of rank, complete or incomplete, and with or without ties. A rank is incomplete if it does not contain all the items. A tie in a rank, in turn, refers to the presence of equally preferred items.

There is a family of initiatives that address rank aggregation from a theoretical perspective of optimal or sub-optimal aggregations. Aledo et al. [3] defined an *extension set* of a rank as the set of permutations that are *compatible* (of equivalent importance) with the given rank, and then proposed a solution for RAP that allows any ranks to be aggregated, based on extension sets. Amodio et al. [5] proposed a heuristic algorithm for RAP that finds one of the existing optimal median ranks in less computational time than more expensive branch-and-bound methods. D'Ambrosio et al. [35] proposed an evolutionary heuristic for RAP, called differential evolution algorithm, that is able to deal with a large number of items in reasonable time, when compared to branch-and-bound and other heuristics. Similarly, Aledo et al. [4] presented evolutionary approaches, and studied the effect of mutation operators, initialization methods, and generation of descendants.

Anyway, RAP is an NP-hard problem for more than three input ranks [48]. In practical search scenarios, rank aggregation can be seen as the task of finding a good permutation of retrieved objects obtained from different input ranks. In this case, rank aggregation methods compose inexact solutions that intend to promote better results than the isolated input ranks. Note that those RAP-based theoretical works have not been explored for retrieval tasks either.

Related to the rank aggregation task, **re-ranking** refers to a family of methods that also intend to promote better results by performing rank repositioning. Re-ranking approaches are feature-based [68] or rank-based [11]. Re-ranking do not explore the inter-relationships between the ranks from the response objects. In this sense, such exploitation is a potential advantage for the rank aggregation methods by definition. Besides, the main advantage of rank-based approaches for improved retrieval, over feature-based approaches, is that while digital objects are typically modeled in high dimensional spaces, they often live in a much lower-dimensional intrinsic manifold space [148]. For this reason, rank-based approaches can be more effective while assuming less input data. *Manifold learning* concerns the exploitation of such intrinsic structure.

Supervised rank aggregation methods are intended to infer fusion formulations automatically from training data, by exploiting labeled information and ground-truth relevance to maximize the effectiveness of a new ranker. Supervised rank aggregation methods belong to the Learning-to-Rank (L2R) field, which comprehends the set of supervised methods for ranking. As a drawback, the availability of training data is not always possible or feasible, and supervised techniques demand more computational cost. A few works have been exploring semi-supervised rank aggregation with some success for image retrieval [38, 109].

Metaheuristic approaches have been proposed, based on Genetic Algorithm (GA), aiming at optimizing the effectiveness of a rank aggregation function for search engines [71, 97]. GA approaches typically apply an optimization process over distance measures to minimize the distances for various aggregated ranks in order to generate a final aggregated rank. Mourão and Magalhães [96], for instance, proposed Learning to Fuse (L2F), a L2R algorithm of presumably lower complexity than other more costly L2R models while achieving competitive retrieval results to them. Their solution mitigates the complexity by analyzing and discarding ranks of minor improvements to the final rank, during its learning process, thus trading precision for complexity.

In our opinion, most supervised rank aggregation models are still either too complex, data-dependent, or costly to scenarios in which unsupervised models can be satisfactory. **Unsupervised rank aggregation** functions work without relying on labeled training data or ground truth information. For that, they can be based on data discrimination or summarization strategies, such as rank position averaging [17, 32], retrieval score combination [52, 104], correlation analysis [107, 147], or clustering [82]. In this work, we focus on unsupervised methods for rank aggregation.

Less frequently, some initiatives have proposed aggregation methods that work upon both object features and ranks. This is a promising approach, but also demands raw data, which may not be available in practical situations. Bhowmik and Ghosh [14] proposed a hybrid unsupervised rank aggregation method that is based on both object attributes

and ranks, as an augmented solution. Unfortunately, their evaluation considered only a few classic baselines (up to 2001) that had not even explored both aspects.

Besides unsupervised or supervised, rank aggregation methods can be also classified as either **order-based** or **score-based**. Score-based methods use the scores associated with each response item from different ranks as input. Order-based methods use only the relative order among the response items to aggregate the ranks.

The first unsupervised works regarding rank aggregation worked solely upon the ranks for a certain query, without exploring the dataset. They were mainly based on position averaging [17, 32, 72] or retrieval score combination [52, 104]. After these approaches, more sophisticated proposals emerged, based on Markov Chains [48, 114], nearest neighborhood and contextual analysis [11, 82, 110, 140], and graph of correlations [107, 147].

BordaCount [17] is a traditional order-based method that computes, for each response item, a new score based on the disparity between its positions on the ranks with respect to the their sizes. Equation 2.2 illustrates how the new scores of each response item x is computed, based on its positions on the ranks $\tau \in \mathcal{T}_q$ for the query q, where $\rho_\tau(x)$ is the position of x in τ.

$$BordaCount(q, x) = \sum_{\tau \in \mathcal{T}_q} (|\tau| - \rho_\tau(x)) \tag{2.2}$$

Reciprocal Rank Fusion (RRF) [32], by contrast, is an order-based method that assigns scores to response items using a formulation that more emphatically penalizes lower-lanker results in favor of highly ranked results. The formula is shown in Equation 2.3, where x is the response item, k is a constant of suggested value 60, $\mathcal{T}_q$ is the set of ranks for the query q, and $\rho_\tau(x)$ is the item position on rank τ.

$$RRF(x) = \sum_{\tau \in \mathcal{T}_q} \frac{1}{\rho_\tau(x) + k} \tag{2.3}$$

Median Rank Aggregation (MRA) [49] is another order-based method. It traverses the ranks counting the number of occurrences of the response items. The first item that occurs in more than half of the ranks is taken as the first item of the final rank. Then, the second item that occurs in more than half of the ranks is taken as the second, and so on.

Six score-based methods were proposed by Fox and Shaw [52]: CombSUM, CombMAX, CombMIN, CombMED, CombMNZ, and CombANZ, based on distinct priors. Table 2.2 indicates how the scores are computed to each object, where $\mathcal{T}_q$ is the set of ranks for the query q, and $\varsigma_\tau(q, x)$ is the score of the object x in the rank τ. For these methods, each rank must be previously normalized with respect to its scores. Related to these methods, RLSim [104] is a score-based technique, inspired by Naive Bayes classifier, that assigns the final score of an object as the product of its scores in each rank.

Condorcet is a voting method, based on the Condorcet criterion. This criterion defines that the winner of the election is the candidate that beats the other candidates in pairwise comparisons. Let the distance between two ranks be the number of pairs whose objects are ranked reversely. The Condorcet winner is the one that minimizes the total distance. The Condorcet method produces a ranking of all candidates from the first to the last

Table 2.2: Scores for the methods by Fox and Shaw [52].

Score	Equation
$CombSUM(q,x)$	$\sum_{\tau \in \mathcal{T}_q} \varsigma_\tau(q,x)$
$CombMAX(q,x)$	$\max_{\tau \in \mathcal{T}_q} \varsigma_\tau(q,x)$
$CombMIN(q,x)$	$\min_{\tau \in \mathcal{T}_q} \varsigma_\tau(q,x)$
$CombMED(q,x)$	$CombSUM(q,x) \div \lvert\mathcal{T}_q\rvert$
$CombMNZ(q,x)$	$CombSUM(q,x) \times \lvert\{\tau : x \in \tau \wedge \tau \in \mathcal{T}_q\}\rvert$
$CombANZ(q,x)$	$CombSUM(q,x) \div \lvert\{\tau : x \in \tau \wedge \tau \in \mathcal{T}_q\}\rvert$

place. The Condorcet winner comes first and the Condorcet loser comes last.

Some **graph-based approaches** for rank fusion were proposed based on Markov Chains, where response items are represented in the various ranks as vertices in a graph, with transition probabilities from vertex to vertex defined by the relative importance of the items in the various ranks [48, 114]. In a similar way, some methods are based on diffusion processes [12, 42], which consists in re-evaluating and re-assigning pairwise affinities – expressed by the edge weights – through the graph in the context of the other objects. This can be done, for instance, by random walks and affinity propagation.

Zhang et al. [147] proposed a graph-based rank aggregation method, referred to here as QueryRankFusion, that explores the notion of reciprocal references. It analyzes the k-reciprocal neighborhoods for building a graph for each rank, and requires the computation of the Jaccard measure for assigning weights to edges. Graphs are later fused into a global graph. Then, it relies on a ranking step using two possible solvers, either based on the PageRank algorithm that computes a transition matrix over the edges, or by a greedy algorithm that finds subgraphs of maximum local density. This method depends on the adjustment of three hyperparameters: the number k of neighbors to analyze; the solver algorithm for the ranking step; and the number of iterations in the ranking step. This method presented effective results, but it was evaluated only for image retrieval tasks.

Similarly, Pedronette et al. [106] proposed RkGraph, a graph-based aggregation approach for distance learning in shape retrieval tasks, which merges graphs defined upon multiple ranks and composes a collection graph.

Pedronette and Torres [105] proposed CorGraph, a learning method based on a correlation graph, which defines the graph connectivity using different levels of correlation measures and exploits strongly connected components. Pedronette et al. [107] continued that previous work by proposing a simpler graph-based method, hereby called RecKNNGraphCCs, as in Zhang et al. [147], but with less intermediate steps and less hyperparameters. In that method, they rely on connected components in the step of generating ranks. A pre-processing step composed of re-ranking and normalization is performed to improve the ranks before the execution of the rank aggregation scheme. This method is affected by two hyperparameters: the number of iterations and the number of neighbors to analyze. This method was also validated only in image retrieval problems.

Yet in graph-based approaches, some works adopt hypergraphs, which consist of a generalization of a graph that allows an edge to link more than two vertices. Hypergraphs

are used in order to capture high-order relationships between objects, rather than limiting relationships between pairs of elements. The approaches typically create a hypergraph from the collection, and then perform an iterative procedure to derive ranks from the analysis of the hypergraph with respect to the query [18, 108, 136]. Bouhlel et al. [18] explored how to fuse too similar objects (represented as vertices) in order to reduce redundancy and improve diversity retrieval. Wang et al. [136] presented an algorithm that combines the adoption of a hypergraph, constructed from textual and visual descriptors, with a relevance feedback procedure to refine the retrieval results.

Existing graph-based methods are mostly targeted at modeling the whole collection of objects as a graph, from which the ranks can be derived. Although this approach can lead to effective retrieval models, this has been still solely restricted to one domain of application. We investigate alternative approaches, for general applicability. To the best of our knowledge, there were no representations based on ranks in the literature so far, not even for single scenarios. This is one objective of our proposed methods.

Another common shortcoming from previous works is about the retrieval scalability, as the query search times are at best asymptotic linear to the collection size [18, 108, 110]. Parallel to those efforts, though, there are initiatives from fields such as database systems regarding **indexing** structures, **embedding** formulations [23, 44, 116], and **approximate search** [89], that can be explored in the context of rank aggregation, although the literature had not yet paid much attention. This is another objective of our proposed methods.

Table 2.3 summarizes the main works regarding unsupervised rank aggregation emphasizing their approaches and limitations. It also highlights our proposed methods in this research field, which are discussed in Chapters 3 and 4. Preliminary works focused on rank fusion without any dataset exploration. They were also focused on textual scenarios only. More recent works started exploring dataset characteristics, but they were mainly restricted or evaluated to image scenarios. Besides, some of them include an computational cost that is not often analyzed or evaluated experimentally. Overall, they also have not yet explored rank-based representations for general applicability.

2.3 Embedding and Indexing

Graph embedding approaches have been effective in multiple scenarios involving graph databases, because vector representations from graphs usually promote better scalability. Also, graph-based embeddings allow the use of existing mining and search functions at disposal for vector representations. An embedding acts as a mapping function from a graph domain to a multidimensional vector space.

Zhu et al. [154] proposed a map that roughly preserves distances between those domains. In their solution, they exploit spaces of high dimensionality. He et al. [63] proposed a learning method for embedding representations of entities and relations previously modeled by graphs. This map can also be based on statistics of vertex attributes and edge attributes [55], prototypes [21], or graph kernels [116]. Among these initiatives, Bag of Graphs (BoG) [116] was introduced as a general unsupervised framework for graph em-

Table 2.3: Main unsupervised rank aggregation functions.

Method	Year	Approach	Unsupervised dataset exploration	Validation scenarios	Efficiency analysis and validation
BordaCount [17]	1781	position averaging	no	none	none
Kemeny [72]	1959	position averaging	no	none	none
CombMAX [52]	1994	score combination	no	text	none
CombMIN [52]	1994	score combination	no	text	none
CombSUM [52]	1994	score combination	no	text	none
CombMED [52]	1994	score combination	no	text	none
CombMNZ [52]	1994	score combination	no	text	none
CombANZ [52]	1994	score combination	no	text	none
Condorcet [95]	2002	position averaging	no	text	analysis
MRA [49]	2003	position averaging	no	text	both
RRF [32]	2009	position averaging	no	text	none
Qin et al. [110]	2011	neighborhood analysis	ranks	image	validation
RLSim [104]	2013	score combination	pairwise distances	image	both
QueryRankFusion [147]	2015	global graph and solver	ranks	image	both
Xie et al. [140]	2015	neighborhood analysis	pairwise distances	image	both
RkGraph [106]	2016	global graph and edge-based distance	ranks	image	none
CorGraph [105]	2016	global graph and connected components	ranks	image / multimodal	none
Bai and Bai [11]	2016	vector of contextual activation	pairwise distances	image	both
RecKNNGraphCCs [107]	2018	global graph and connected components	ranks	image	none
Liang et al. [82]	2018	clustering	pairwise distances	text	both
Bai et al. [12]	2018	diffusion process	ranks	image / multimodal	validation
FG (Ours) [45]	2019	graph-based representation	ranks	text / image / multimodal	both
Bouhlel et al. [18]	2020	clustering & hypergraphs	ranks	multimodal	none
FV (Ours) [43]	2019 – 2020	graph-based representation embeddings	ranks	text / image / multimodal	both

bedding that allows graphs to be represented as vectors based on common local graph patterns. Despite BoG is targeted for any graph scenario, it requires some functions to explicitly defined concerning the target scenario. BoG has been extended for some scenarios already, such as for text classification and retrieval [44].

Complementary, indexing mechanisms have been extensively studied in the information retrieval literature, aiming at performing query retrieval efficiently by means of either exact or approximate nearest neighborhood search. These solutions usually adopt space partitioning [90], hashing [56], or greedy search in neighborhood graphs [89].

We claim that graph embedding and indexing strategies can be applied for graph-based representations for rank aggregation, in order to promote efficiency capabilities and more generic representations. We investigate these novel paradigms in the next chapters.

2.4 Data Fusion

Representation models and learning systems have been developed and advanced for the data modalities individually, such as image, text, video, and audio. However, the exploration of multiple data sources and descriptors combined is still an open issue. Multimodal tasks may impose even more challenges, depending on the scenario, such as translation between modalities, exploration of complementarity and redundancy, co-learning, and semantic alignment [13].

Some works focused on multimodal events, which require modeling spatio-temporal characteristics of data [128]. Faria et al. [50] proposed a time-series descriptor that generates recurrence plots for series, coupled with a bio-inspired optimization focused on combination of classifiers. We focus on multimodal tasks that do not depend on temporal modeling as well as unsupervised models.

In order to achieve fusion capabilities, *early-fusion* approaches emphasize the generation of composite descriptions for samples, thus working at feature level. Conversely, *late-fusion* approaches perform a combination of techniques focused on a target problem, fusing at score or decision level. On a smaller scale, some papers propose hybrid solutions based on both approaches [78, 145].

While early-fusion approaches are theoretically able to capture correlations between modalities, often a certain modality produces unsatisfactory performance and leads to biased or over-fitted models [145]. Most early-fusion methods work in a two-step procedure, first extracting features from different modalities, then fusing them by strategies, such as concatenation [73], singular values decomposition [20], or autoencoders [115]. A few other methods focus on multimodal features jointly [65, 75], although generally restricted to a pre-defined textual attribute set. Concatenation is a straightforward yet widely used early-fusion approach, which merges vectors obtained by different descriptors. As a drawback, concatenation does not explore inherent correlations between modalities.

Supervised early-fusion optimizes a weighted feature combination, either during or after feature extraction. A common strategy is to build a neural architecture with multiple separate input layers, then including a final supervised layer such as a regressor [100]. Another approach is to design a composite loss function, suited for the particular de-

sired task [102]. Composite loss functions work well in practice, but, as they need both multimodal composition and supervision, they are tied to the domain of interest. Supervised early-fusion usually suffers from high memory and time consumption costs. Besides, they usually have difficulty in preserving feature-based similarities and semantic correlations [78].

Late-fusion approaches are particularly useful when the raw data from the objects are not available. Besides, they are less prone to over-fit. Mixture of experts (MoE) approaches focus on performing decision fusion, combining predictors to address a supervised learning problem [144]. Majority voting of classifiers [100], rank aggregation functions [82, 147], and matrix factorization [41] are examples of late-fusion methods. Both fusions based on rank aggregation functions and matrix factorization are based on manifold learning, i.e., the exploration of intrinsic dataset geometry.

Majority voting is a well-known approach to combine multiple estimators, being effective due to bias reduction. It is applied to scenarios involving an odd number of estimators, so that each predicted output is taken as the one most frequently predicted by the base estimators.

Regarding previous works on fusion approaches for prediction tasks, a considerable amount of them are still based on classic visual descriptors [37, 60, 125, 137]. Most of them resort to pre-trained CNN-based models for visual feature extraction [1, 2, 7, 15, 54, 86, 99], from which just a few fine-tune their models [86, 99]. When dealing with specific tasks, specially during annual competitions, some works use preprocessing steps, such as image cropping and filtering [149], but this is beyond our general intent in this work, regarding fusion methods and representation models.

In order to explore the textual modality, most initiatives use Bag of Words (BoW), using either Term Frequency (TF) or Term-Frequency – Inverse Document Frequency (TF-IDF) weighting, while others present more complex formulations, such as word embeddings [15, 125], Long Short-Term Memory networks [86] (LSTMs), or relation networks [99].

Regarding multimodal scenarios, most works rely on early-fusion approaches, such as a concatenation of visual and textual feature vectors [15, 37, 54, 60, 86, 125] or graph-based attribute fusion [7, 137], while only a few others adopt late-fusion approaches [1, 2].

2.5 Representation Learning and Feature Selection

Related to representation models, **representation learning** is a research field that tries to optimize feature generation and feature selection in order to obtain a good feature set, which in turn must ideally be *concise*, *representative*, and *discriminative*. Conciseness here refers to a small yet enough size. Representative features are those that generalize well, and a discriminative feature is one different enough from the rest. These two steps are complementary. Feature generation extracts feature candidates, which are then analyzed and ranked by a feature selection protocol according to their (presumed) discriminative power. They can be either performed one after another, or together by a unified learning process.

Representation learning is critical for any desired task, because a good feature set is expected to better approximate samples from the same class or subject. By doing so, it should increase the probability of a test sample to be correctly classified, or increase the effectiveness of a retrieved rank in a retrieval system. In our opinion, representation learning can benefit embedding approaches, which are commonly unsupervised, for preserving either properties or distances from the graph domain to the vector domain, while optimizing the feature set generation. We are particularly interested in evaluating how representation learning may benefit the embeddings of graph-based representation models from fusions of ranks.

Feature selection approaches have been proposed by several strategies, such as feature clustering [30, 33], unsupervised distance learning [118, 143], feature importance estimation [10, 51, 133], a posteriori feature analysis and filtering [34, 88], inner representation of pre-trained deep neural networks [142], evolutionary optimization heuristics [24, 120], or end-to-end learning [29, 59, 85]. Complementary, there are also a vast number of unsupervised and supervised weighting schemes. Some of them were originally designed for text mining, although they can be applied in other domains as well. Document Frequency (DF), Term Frequency (TF), Term-Frequency – Inverse Document Frequency (TF-IDF) [9], and *Support* are examples of unsupervised functions. Information Gain (IG) [94], Mutual Information (MI) [141], and Chi-Square (χ^2) [119] are examples of supervised functions.

Let $DF(f_i)$ be the number of samples from the corpus that a feature f_i appear, and $TF(f_i, s_j)$ be the proportion rate of occurrences of f_i in the document (sample s_i). DF and IDF are commonly used (i) in combination, (ii) as preliminary filters along with thresholds, or (iii) as components of more complex weighting functions.

Support refers to a threshold of minimum DF score that a feature has to achieve to be maintained in the feature set, otherwise it is discarded. TF-IDF is a commonly used weighting approach in text classification. It measures the relative frequency of terms (f_i) in a specific document (sample s_j) through an inverse proportion of the term over the whole corpus of size N. TF-IDF is given by Equations 2.4 and 2.5, where Inverse Document Frequency (IDF) is adopted to estimate the importance of f_i in the corpus, such that a feature of high IDF appears in few samples.

$$IDF(f_i) = log(\frac{N}{DF(f_i)}) \tag{2.4}$$

$$TF - IDF(f_i, s_j) = TF(f_i, s_j) \times IDF(f_i) \tag{2.5}$$

χ^2 is a statistical measure that follows the intuition that the best features f_i for the class c_j are the ones distributed most differently in the sets of positive and negative samples of class c_j. It measures the association between a feature and a class. High χ^2 scores indicate that the occurrence of the feature and the class are statistically dependent. χ^2 of a feature f_i is defined by Equations 2.6 and 2.7, where (i) A and B are the number of samples, in class c_j, that respectively contain f_i and do not contain f_i, (ii) C and D are the number of samples, not in class c_j, that respectively contain f_i and do not contain f_i, and (iii) N is total number of samples in the training corpus, such that $N = A+B+C+D$.

$$\chi^2(f_i, c_j) = \frac{N(AD - BC)^2}{(A + C)(B + D)(A + B)(C + D)}. \tag{2.6}$$

$$\chi^2(f_i) = \max_{1 \leq j \leq |c|} (\chi^2(f_i, c_j)). \tag{2.7}$$

Both unsupervised and supervised feature selection methods are promising in many scenarios, but they also pose limitations. In MI, for example, rare features have a higher score than common features [61]. In χ^2, the features selected (those top scored ones) do not follow any distribution guidance: the feature distribution per class do not follow the proportion of the number of samples per class [10]. Therefore, the classification by means of those features could be impacted.

The class frequency of a feature f, expressed as $c(f)$, is the number of classes that present any sample containing f. In [133], a score called relevance class frequency (RCF) was defined for term pairs. The main idea is to prioritize the selection of a term pair whose composite class frequency is small, relative to its subfeatures. By doing so, such composite features would be more discriminant and less redundant. The RCF for a feature pair f_{ij} is given by Equation 2.8. $RCF(f_{ij})$ varies from 1 to $\min(c(f_i), c(f_j))$ because f_{ij} is derived from f_i and f_j, so $c(f_{ij})$ is always less or equal than $c(f_i)$ and $c(f_j)$. Besides, for the same reason we say that a feature whose RCF is 1 is redundant.

$$RCF\left(f_{ij}\right) = \frac{\min\left(c\left(f_i\right), c\left(f_j\right)\right)}{c\left(f_{ij}\right)} \tag{2.8}$$

Chapter 3

Graph-based Rank Aggregation and its applications in Retrieval Tasks

3.1 Introduction

In this chapter, we propose a novel unsupervised graph-based rank aggregation function, agnostic of the rankers being fused, and targeted for general applicability, such as image, textual, or even multimodal retrieval tasks. As we summarized in Chapter 2, existing graph-based methods for rank aggregation functions have been the most successful, although they are mostly concerned at modeling the whole collection of objects as one graph, from which the ranks can be derived. Different from previous works, we model one graph per object, and redefine the object retrieval system by means of those graphs. We reformulate the ad-hoc retrieval problem as a document retrieval based on *fusion graphs*, which we propose as a new unified representation model capable of merging multiple ranks and express inter-relationships of retrieval results automatically.

We investigate how the retrieval system can benefit from learning the manifold structure of datasets, thus promoting more effective results. We also evaluate the impact of different ranker selection criteria for fusion, which take into account the rankers' effectiveness and their correlation.

Our approach is more robust for some reasons. First, our graph definition not only encapsulates the information from ranks of a certain query, but also incorporates information regarding inter-relationships between the result items and their own ranks, which is not fully performed by some other methods. Second, our aggregation approach can actually serve not only to establish a retrieval model from a rank aggregation function, but also it can be suitable as a preliminary representation for any multimodal task. Third, the proposed method does not require free parameters, such as neighborhood size definition for the graph construction. In does not require time-consuming tuning of hyperparameters.

The proposed method is also innovative with regard to the definition of the fused retrieval score. While other related graph-based approaches exploit the graph through operations on transition matrices [147] or specific similarity measures [107], our approach derives a new retrieval score directly based on the graph structure, considering the minimum common subgraph of two objects' graphs. In summary, our ranking system relies on

exploiting contextual information obtained from the direct comparison of objects based on their neighbors, which are defined in terms of the ranks associated with multiple ranking criteria, in unsupervised manner.

Our approach presents theoretical and practical advances and implications. A theoretical implication is that ranks can be directly used for fusion, thus promoting a unified representation. From this representation, called fusion graph, we derive a straightforward ranking procedure, without further transformations and optimizations. That corresponds to the second theoretical implication. Finally, a practical implication from our approach is its advantage that the fusion graph extraction and the graph-based retrieval are independent, both being capable of adaptation or further improvement. In addition, our solution does not require time-consuming tuning of hyperparameters.

In Section 3.3, we provide a comprehensive experimental evaluation that was conducted comprising search tasks over six datasets of diverse purposes, three scenarios for each dataset, and state-of-the-art baselines. In Section 3.4, we summarize our findings and propose future research directions.

This chapter advances the literature in terms of the following contributions:

1. The proposal of a novel graph-based rank aggregation model,

    - which is unsupervised, does not require tuning of hyperparameters, and yields top performance compared to state-of-the-art baselines and large gains over the rankers being fused;
    - which is agnostic about the ranks, such as how they are generated, their weighting functions, or whether they are based on distance or similarity scores;
    - which is flexible as its components, the fusion graph extraction and the graph-based retrieval, are independent, both being capable of adaptation or further improvement.

2. The proposal of *fusion graphs*, a graph representation, which is capable of merging multiple ranks and expressing inter-relationships of retrieval results automatically. The proposed representation intrinsically supports multimodal objects, meaning that it can be applied over ranks defined according to different data types at same time;

3. Unlike existing approaches, a straightforward ranking procedure is proposed, for the fusion representation. The method does not require optimizations or additional processing steps;

4. A novel similarity score is formulated, based on the fusion graphs, using an efficient computation of minimum common subgraphs.

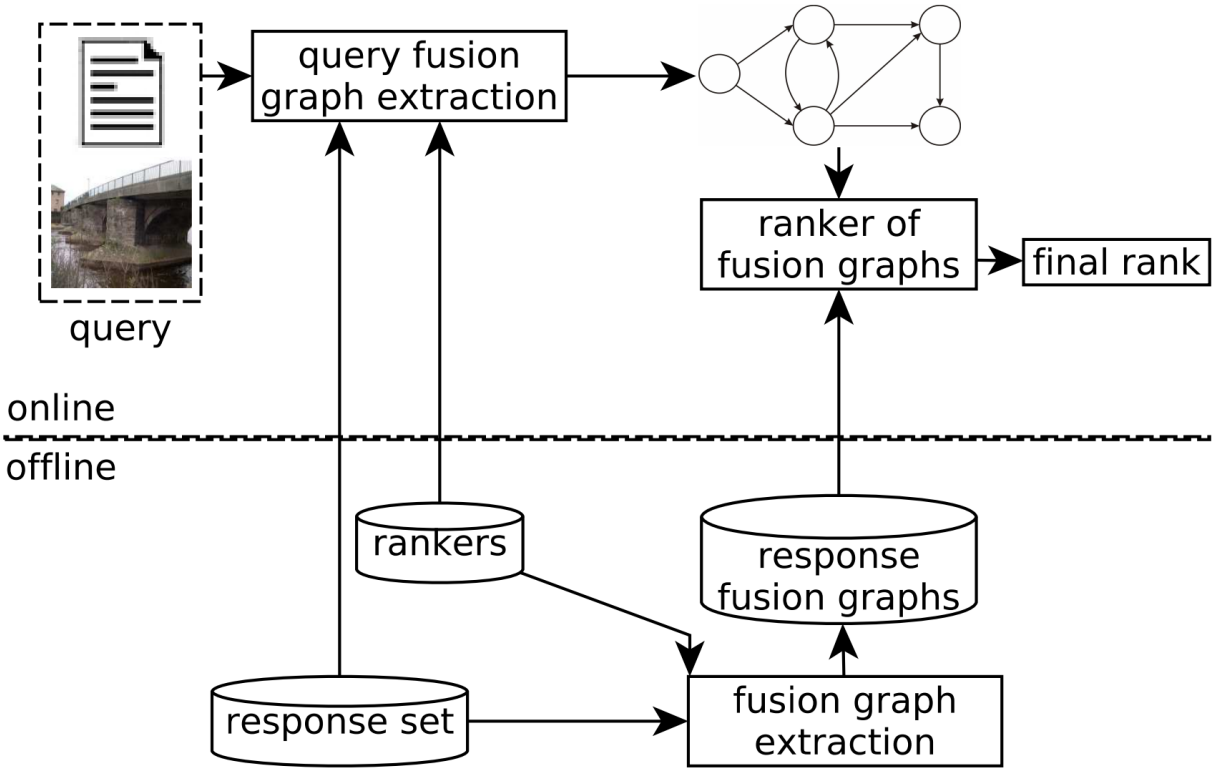


Figure 3.1: Schematic view of the unsupervised graph-based rank aggregation approach.

3.2 Unsupervised Graph-based Rank Aggregation Approach

We propose a graph-based rank aggregation function f that works for any collection S combined with the use of m rankers of any kind. It relies on a *composite* ranker, whose descriptor extracts a graph-based representation, named *fusion graph*, from collection samples, and a fusion graph comparator is employed in this ranker. A fusion graph encodes contextual information from different ranks, defined in terms of multiple base rankers.

Both a query q and each sample s of a target collection are represented by graphs, say query fusion graph $G_{\mathcal{T}_q}$ and fusion graph $G_{\mathcal{T}_s}$. A search is, therefore, modeled as the ranking of graphs $G_{\mathcal{T}_s}$ of collection samples with respect to a query graph $G_{\mathcal{T}_q}$, i.e., the rank aggregation function f is able to rank fusion graphs based on their similarity to a query graph.

Figure 3.1 provides a schematic overview of the unsupervised graph-based rank aggregation approach, which is composed of offline and online workflows. The steps 'fusion graph extraction' and 'ranker of fusion graphs' are detailed in Sections 3.2.1 and 3.2.2, respectively. The offline workflow is responsible for representing the response set as fusion graphs, while the online workflow, in turn, processes a query and produces a final rank to be returned as the final result.

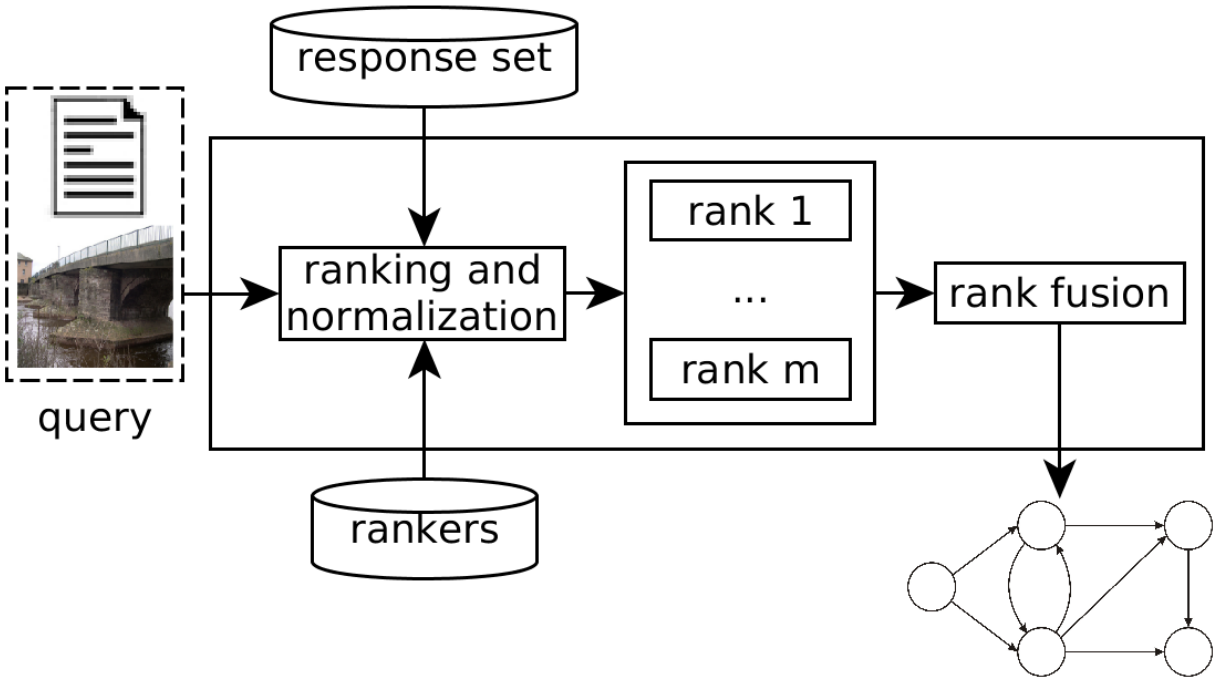


Figure 3.2: Extraction of a fusion graph.

3.2.1 Extraction of Fusion Graphs

In both offline and online workflows, illustrated in Figure 3.1, a fusion graph extraction step is adopted. A fusion graph extraction aggregates ranks for a query, based on the rankers and response set used, producing a fusion graph per query. It is basically comprised of three steps: creation of ranks using different rankers, rank normalization, and rank fusion. These components are illustrated in Figure 3.2. The generation of ranks follows the definitions given in Section 2.1. Our rank aggregation function works upon a predefined set of rankers, so we assume that the base ranks for any query can be provided as requested. The following sub-sections and detail the other steps.

Rank Normalization

For a certain ranker, its comparator $\mathcal{C}$ may be either a distance or similarity function. Furthermore, different comparators may produce scores at different ranges. Nevertheless, these scores are employed in our rank aggregation function. For this reason, we need to normalize the comparator outputs, from the rankers being used, so that the scores from ranks become comparable. The ranks' scores must also fit into an uniform positive range, due to the way we use the scores in our rank aggregation formulation.

Assuming ranks of size L, we adopt a rank normalization procedure that relies on two steps: rank repositioning based on mutual and reciprocal rank references, and score rescaling.

Rank relationships are not symmetric, in the sense that an object i well ranked for a query j does not imply that j is well ranked for a query i. However, improving the symmetry of the k-neighborhood usually improves the effectiveness of retrieval functions [69]. In order to explore this behavior, we propose a rank repositioning, based on a neighborhood-

aware distance δ given by Equation 3.1, where $\rho_{\tau_i}(j) \leq L$ and refers to the position of j in the rank τ_i. It considers both mutual [104] and reciprocal [110] neighborhood, and the ranks are then updated by a stable sorting algorithm over δ, up to the top-L positions. The idea is to bring a ranked item i to the top positions of the rank of j as much as j also has i in top positions of its own rank. The mutual neighborhood sums rank positions from both ranks, and the reciprocal neighborhood considers only the maximum.

$$\delta(i,j) = \rho_{\tau_i}(j) + \rho_{\tau_j}(i) + max(\rho_{\tau_i}(j), \rho_{\tau_j}(i)) \tag{3.1}$$

For the second rank normalization step, we perform score re-scaling for the rank, assigning a uniform range from 1, to the top-ranked response item, to 0.1, to the top-L ranked response item, adopting uniform steps within this interval. We adopt 0.1 as the minimum score because the weights have to be positive, in our rank fusion step. Although, at first, it might be reasonable to just re-scale the rank scores to a uniform interval, such as $[0.1, 1]$, which indeed does not affect the statistical distribution of the original data (as it is a translation followed by a multiplicative scaling), we noticed empirically that our approach is more robust to the presence of outliers and heterogeneous rankers.

Rank Fusion

This step is responsible for producing graphs for query samples that reflect their ranks, and the relationships between their ranks. At first, in an offline stage, for each sample $s \in S$, we perform a search using s as q and obtain its corresponding set $\mathcal{T}_q$ of ranks, using a cut-off of L.

The choice of L depends on the intended result size. Due to the way we construct the fusion graph, especially the vertex and edge weights, the value of L is not supposed to affect the quality of the model, thus not demanding empirical adjustment. The effect of changing the value of L is to increase the effectiveness of the method, up to a certain limit. In practice, the choice of L is guided only by the trade-off between efficiency and effectiveness. Our method has only this parameter, as opposed to some related works [107, 147], usually dependent on multiple hyperparameters.

From $\mathcal{T}$, we derive a weighted directed graph $G_{\mathcal{T}} = (V_{\mathcal{T}}, E_{\mathcal{T}})$ that combines information from the ranks of $\mathcal{T}$, where $V_{\mathcal{T}}$ is the vertex set and $E_{\mathcal{T}}$ is the edge set. A fusion graph aims to be a discriminative and comparable representation of objects, based on their ranks and existing relations among ranks. In this way, a fusion graph $G_{\mathcal{T}_q}$ of an object q includes all response items from each rank $\tau_q \in \mathcal{T}_q$, as vertices. Vertices are connected by taking into account the degree of relationship between them, and the degree of their relationships to q.

Algorithm 3.1 illustrates how $G_{\mathcal{T}}$ is computed. A vertex v_A is associated with a collection sample A. The vertex set is composed of the union of all samples found in all ranks defined for query q. The weight of vertex v_A, w_{v_A}, is the sum of the similarity similarities that the response item A has in the ranks of q (lines 5 to 10, Equation 3.2). The vertex weight is expected to encode how relevant a response item A is to q.

Edges are created to express the relationship between response items (lines 11 to 20). There will be an edge $e_{A,B}$, linking v_A to v_B, if A and B are both responses in any rank of

Algorithm 3.1: Rank fusion.

```
1  # inputs: ranks T_q, for the query q
2  # output: a weighted directed graph G_T
3  G_T = WeightedDirectedGraph() # (V_T, E_T)
4  for τ in T_q: # create vertices
5    for A in τ:
6      weight = ς_τ(q, A)
7      if v_A ∉ V_T: # if new vertex
8        V_T = V_T ∪ v_A
9        w_vA = weight
10     else: w_vA += weight
11 for τ in T_q: # create edges
12   for A in τ:
13     for τ_A in T_A:
14       for B in τ_A:
15         if v_B ∈ V_T and A != B:
16           weight = ς_τA(A, B) ÷ ρ_τ(A)
17           if e_A,B ∉ E_T: # if new edge
18             E_T = E_T ∪ e_A,B
19             w_eA,B = weight
20           else: w_eA,B += weight
21 g.normalizeWeights(0,1)
```

q and if B occurs in any rank of A. The weight of $e_{A,B}$, $w_{e_{A,B}}$, is the sum of the similarities that the response item B has in the ranks of A, divided by the position of A in each rank of q (Equation 3.3), considering position values starting by 1. The scores of B in each τ_A matters, so we sum them. Also, we weight these scores inversely to the position in which A appears in τ_q. The goal is to ensure that the weight of the edge between A and B also encodes the importance of A with respect to q.

$$w_{v_A} = \sum_{A \in \tau_i \wedge \tau_i \in \mathcal{T}_q} \varsigma_{\tau_i}(q, A) \tag{3.2}$$

$$w_{e_{A,B}} = \sum_{A \in \tau_i \wedge \tau_i \in \mathcal{T}_q} \sum_{B \in \tau_j \wedge \tau_j \in \mathcal{T}_A} \left(\varsigma_{\tau_j}(A, B) \div \rho_{\tau_i}(A)\right) \tag{3.3}$$

The creation of $G_{\mathcal{T}}$ ends with a weight normalization (line 21), which makes the graph comparable by means of a graph comparator. The weight of each vertex v_i, w_{v_i}, is replaced by $\frac{w_{v_i}}{max(w_v)}$, and the weight of edge $e_{i,j}$, $w_{e_{i,j}}$, is replaced by $\frac{w_{e_{i,j}}}{max(w_e)}$.

Figure 3.3 illustrates an example for the rank fusion, assuming a query q and the use of two rankers. At first, a fully disconnected graph is built based on the retrieved results and their scores (I). Then, the relationships between the results (encoded in the ranks for B in blue, C in green, and D in orange), in their own ranks, are propagated into the graph as edges (see III, IV, and V). The resulting graphs, VI and VII in Figure 3.3, correspond to the final fusion graph before and after normalization, respectively.

$\tau_1 q$ = ((A,1), (B,0.6), (C,0.1))
$\tau_2 q$ = ((B,1), (C,0.3), (D,0.1))

$\tau_1 A$ = ((B,1), (D,0.4), (E,0.1))
$\tau_2 A$ = ((B,1), (E,0.5), (F,0.1))
$\tau_1 B$ = ((D,1), (A,0.9), (E,0.1))
$\tau_2 B$ = ((A,1), (C,0.2), (D,0.1))
$\tau_1 C$ = ((D,1), (A,0.4), (F,0.1))
$\tau_2 C$ = ((E,1), (F,0.7), (A,0.1))
$\tau_1 D$ = ((B,1), (F,0.9), (E,0.1))
$\tau_2 D$ = ((C,1), (E,0.6), (B,0.1))

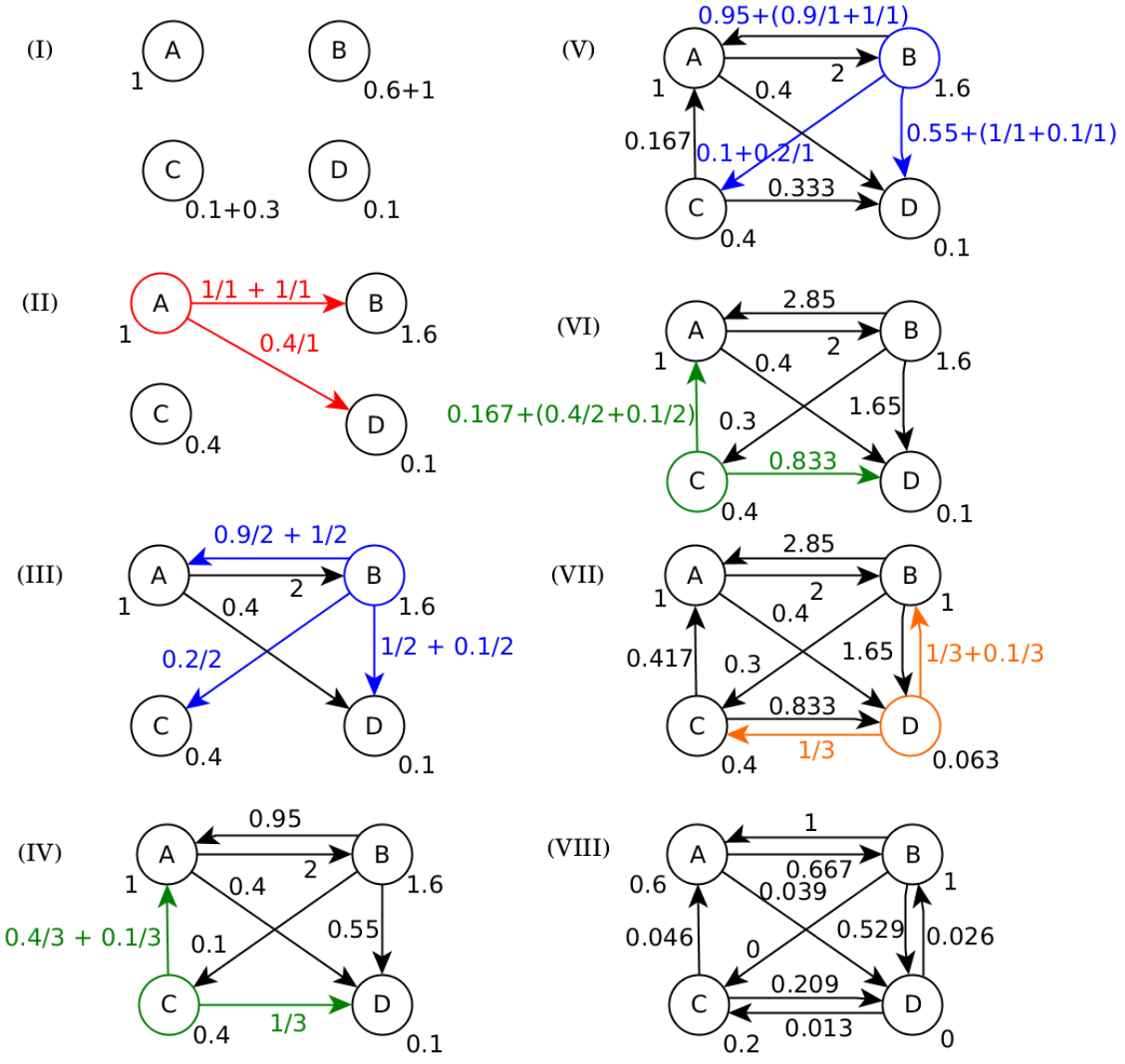


Figure 3.3: Example of graph construction during rank fusion. The scores are shown along with the response items, within the ranks, for simplicity.

3.2.2 Retrieval based on Fusion Graphs

Our proposed rank aggregation function f relies on a *composite* ranker that is defined as $R_G = (D_G, \mathcal{C}_G)$, where D_G is a descriptor which extracts fusion graphs, and $\mathcal{C}_G$ is a fusion graph comparator. Given two fusion graphs G_a and G_b, $\varsigma(G_a, G_b)$ can be computed by $\mathcal{C}_G$ using any graph-based similarity or dissimilarity function. We propose the adoption of formulations based on the minimum common subgraph (mcs), such as MCS [22] or WGU [132]. A graph M is the mcs of two weighted graphs G_a and G_b if: (1) $M \subseteq G_a$ (2) $M \subseteq G_b$ and (3) there is no other subgraph M' ($M' \subseteq G_a$, $M' \subseteq G_b$), such that $|M'| > |M|$, where $|M|$ is given by the sum of the vertex weights and edge weights of M.

MCS and WGU are shown in Equation 3.4 and 3.5, respectively. In MCS, the larger the $|mcs|$ is, the more similar the two graphs are, which decreases the distance up to 0. This metric produces values in [0, 1]. WGU behaves similarly to MCS with respect to identical graphs or graphs without intersection, and also produces values in [0, 1]. The denominator in WGU represents the size of the union of the two graphs, whose motivation is to allow for changes in the smaller graph to influence the distance value, which is not covered in MCS.

$$dist_{MCS(G_a,G_b)} = 1 - \frac{|mcs(G_a, G_b)|}{max(|G_a|, |G_b|)} \tag{3.4}$$

$$dist_{WGU(G_a,G_b)} = 1 - \frac{|mcs(G_a, G_b)|}{|G_a| + |G_b| - |mcs(G_a, G_b)|} \tag{3.5}$$

Note that the scores $\varsigma(G_{\mathcal{T}_q}, G_{\mathcal{T}_{s_i}})$ and $\varsigma(G_{\mathcal{T}_q}, G_{\mathcal{T}_{s_j}})$ can be compared to infer whether s_i or s_j is more relevant to q. The higher the score, the most similar the query and the response item are, regarding their ranks. A fusion graph, therefore, is able to encode intrinsic contextual information from multiple ranks.

The rank aggregation function f is defined as $f(\mathcal{T}_q) = \tau_{q,f} = \{s_1, s_2, \ldots, s_n\}$ such that $|\tau_{q,f}| \leq L$ and $\{\varsigma(G_{\mathcal{T}_q}, G_{\mathcal{T}_{s_1}}), \varsigma(G_{\mathcal{T}_q}, G_{\mathcal{T}_{s_2}}), \ldots, \varsigma(G_{\mathcal{T}_q}, G_{\mathcal{T}_{s_n}})\}$ is in increasing order.

3.2.3 Computational Cost Analysis

The base rankers typically adopt indexing structures, such as KD-tree or inverted files, leading to rank generations in sub-linear time with respect to the response set, taking $O(\log n)$. One rank is stored in $O(L)$, due to our hyper-parameter L.

The rank fusion (Algorithm 3.1) has an asymptotic cost of $O(mL)$ for the first outer loop (create vertices), and $O(mLmL)$ for the second outer loop (create edges), leading to a total cost of $O(m^2L^2)$. As we use small values of L and also a small number (m) of rankers, the cost of the rank fusion algorithm itself is negligible when compared to the step concerned with the generation of ranks. The number of vertices in each fusion graph is $O(mL)$ in the worst case. Therefore, the storage of a fusion graph is $O(m^2L^2)$ in the worst case.

In our fusion graph, vertices have different labels, and for this reason we benefit from using graph comparison functions that take advantage of efficient algorithms for computing minimum common subgraphs, reducing its asymptotic cost to $O(|V_1||V_2|)$ [40]. The comparison between two fusion graphs is bounded to $O(m^2L^2)$. Both aspects lead to

efficient graph comparators, in practice.

The search, for each q, relies on the existence of ranks and fusion graphs from the response set, but both steps are performed only once per collection ('fusion graph extraction' in Figure 3.1), in an offline stage. The rank generation for the response set, considering m rankers, takes $O(n(m \log n))$, and requires $O(n(mL))$ for storage. As for the response fusion graphs, it takes $O(nm^2L^2)$ for both generation and storage. The final cost of the offine stage is $O(n(m \log n + m^2L^2))$ for execution, and $O(nm^2L^2)$ for storage, or approximately $O(n \log n)$ and $O(n)$ for small values of m and L, which is overall efficient.

The cost of executing one query is the sum of the costs for: generating the ranks $\mathcal{T}_q$ for q; generating the query fusion graph $G_{\mathcal{T}_q}$; and retrieving samples using $G_{\mathcal{T}_q}$ and response set fusion graphs. For the first part, the generation of individual ranks $\tau \in \mathcal{T}_q$ is asymptotically limited by the cost of the slowest ranker. In general, this part takes $O(m \log n)$. For the second part, it takes $O(m^2L^2)$. For the third part, the graph-based retrieval can be either implemented linearly over the response set, or sub-linearly using indexing methods such as graph embedding techniques [44, 116, 154]. Each full search over the response set takes $O(n(m^2L^2)) \approx O(n)$. We leave the investigation of sub-linear search in this context for future work.

3.3 Experimental Evaluation

This section presents the adopted evaluation protocol and experimental results related to the comparison of our method with individual rankers and other rank aggregation approaches.

3.3.1 Datasets and Features

We selected datasets of different purposes, compositions, and sizes in order to validate our method in different searching scenarios. For each dataset, we enumerate a few promising heterogeneous descriptors for exploration, but many others could be adopted. As we are more focused on evaluating the aggregation aspects, the choice of descriptors and rankers is not the main intent here. If desired, one may initially select rankers per dataset in order to get those most accurate and less correlated. Table 3.1 lists the datasets used, and Table 3.2 summarizes the individual rankers being adopted per dataset in order to (1) be evaluated in isolation, and (2) generate rankers for the fusions.

Ohsumed [64] is a textual dataset, composed of bibliographic medical documents, provided by the National Library of Medicine. It contains 34,389 cardiovascular diseases abstracts, distributed across 23 Medical Subject Headings (MeSH) diseases categories of cardiovascular diseases group. Without loss of generality, we used the subset of 18,302 uni-labeled documents, varying from 56 to 2876 documents per category. For Ohsumed, we adopted 7 rankers[1]:

[1]For all textual rankers used in the experimental evaluation, we preprocess the documents with stop word removal and Porter's stemming.

Table 3.1: Datasets used in the experimental evaluation.

Dataset	Size	Type
Ohsumed	34,389	Textual
Brodatz	1,776	Texture
MPEG-7	1,400	Shape
Soccer	280	Color Scenes
UW	1,109	Color Scenes and Keywords
UKBench	10,200	Objects / Scenes

Table 3.2: Individual rankers adopted per dataset in the experimental evaluation.

Dataset	Rankers	Type
Ohsumed	BoW-cosine, BoW-Jaccard, 2grams-cosine, 2grams-Jaccard, GNF-MCS, GNF-WGU, WMD	Textual
Brodatz	LBP, CCOM, LAS	Texture
MPEG-7	SS, BAS, IDSC, CFD, ASC, AIR	Shape
Soccer	GCH, ACC, BIC	Color
UW	GCH, BIC, JAC, HTD, QCCH, LAS, COSINE, JACCARD, TF-IDF, DICE, OKAPI, BOW	Color, Texture, Textual
UKBench	ACC, VOC, SCD, JCD, CNN-Caffe, FCTH, CEDD	Color, Texture, BoVW, CNN

- 2 using the BoW, with comparators cosine and Jaccard: BoW-cosine and BoW-Jaccard;
- 2 using the 2grams descriptor, with comparators cosine and Jaccard; 2grams-cosine and 2grams-Jaccard;
- 2 using a graph-based descriptor, called *normalized-frequency* (GNF) [113], with comparators MCS [22] and WGU [132]: GNF-MCS and GNF-WGU;
- WMD [77], a ranker based on *word embeddings* [92].

Brodatz [19] is a texture dataset. There are 1,776 images (texture blocks), being 16 samples for each of the 111 classes (texture types). We adopt 3 texture rankers: Local Binary Patterns [101] (LBP), Color Co-Occurrence Matrix [76] (CCOM), and Local Activity Spectrum [123] (LAS).

MPEG-7 [79] is a shape dataset, composed of 1400 images, equally distributed in 20 images per 70 categories. We adopt 6 shape rankers: Segment Saliences [127] (SS), Beam Angle Statistics [6] (BAS), Inner Distance Shape Context [83] (IDSC), Contour Features Descriptor [103] (CFD), Aspect Shape Context [84] (ASC), and Articulation-Invariant Representation [57] (AIR).

Soccer [130] is an image dataset, composed of 280 images, equally distributed in 40 images per 7 categories (the soccer teams). We adopt 3 color-based rankers: Global Color Histogram [122] (GCH), Auto Color Correlogram [67] (ACC), and Border/Interior Pixel Classification [121] (BIC).

University of Washington (**UW**) [39] is a hybrid dataset, composed of 1109 pictures from different locations, annotated by textual keywords. The number of keywords per picture vary from 1 to 22. There are 20 classes, varying from 22 to 255 pictures per class. We adopt 12 rankers, comprising 3 types:

- 3 Visual color rankers: GCH, BIC, and Joint Autocorrelogram [138] (JAC);
- 3 Visual texture rankers: Homogeneous Texture Descriptor [139] (HTD), Quantized Compound Change Histogram [66] (QCCH), and LAS;
- 6 Textual rankers: COSINE [9], JACCARD [81], TF-IDF [9], DICE [81], OKAPI [111], and BOW [25].

UKBench [98] is a dataset of 10,200 images, consisting of 2,550 scenes/objects captured 4 times each. The captures vary in terms of viewpoint, illumination, and distance. The objects/scenes correspond to the categories, so there are four samples per class. Due to the small and fixed category sizes, effectiveness assessment using this dataset relies on an evaluation metric, called N-S Score, varying from 1 to 4, which measures the mean number of relevant images among the first four images retrieved. We adopt seven rankers, based on color and texture properties. Some of them are based on global descriptors, while others rely on local features:

- ACC;
- Vocabulary Tree [135] (VOC), that uses SIFT;
- CNN-Caffe [70]: features extracted from the 7th layer of a Convolutional Neural Network (CNN) obtained with the Caffe framework. A 4096-dimensional descriptor is extracted per image, and the Euclidean distance is used as the comparator.
- Scalable Color Descriptor [91] (SCD)
- Joint Composite Descriptor [146] (JCD)
- Fuzzy Color and Texture Histogram Spatial Pyramid [28] (FCTH)
- Color and Edge Directivity Descriptor Spatial Pyramid [27] (CEDD)

3.3.2 Experimental Procedure

We evaluate our method, as well as the individual rankers and baselines, with respect to the effectiveness in retrieval tasks. For the Ohsumed dataset, we implemented the rankers and extracted the ranks ourselves. For the other datasets, we adopted ranks built from previous works of our research group [105, 107].

Due to the nature of the datasets used, we use each sample s as query q at a time, whose result candidates belong to S, and we consider a retrieved item as relevant to the query if it belongs to the same class of the query sample, since we are validating in labeled collections, i.e., relevant labels in the experiments are either 1 for relevant or 0 for irrelevant. Therefore, in this case, the query set size corresponds to the dataset size.

Separate query and response sets can be used, as well as graded relevance, but these aspects do not affect the applicability of our model. This protocol concerns document retrieval, also referred to as ad hoc retrieval, which was also very usual in validation protocol of our baselines. We use Normalized Discounted Cumulative Gain at cutoff 10 (NDCG@10) for all datasets except for UKBench, for which we use the adopted N-S Score effectiveness measure, the standard measure used in this dataset.

NDCG is an overall good metric for some reasons. First, a diverse range of metrics has been used in the literature for the same datasets without specific reasons, which reduces their uniformity and reproducibility. Moreover, NDCG is preferred over metrics such as Precision (P), Recall (R), MAP or Bull's Eye Score (R@40), because: it takes into account graded relevance; it analyzes relevance weighted by rank positions; and does not require the computation of full ranks, such as MAP does. NDCG is a normalized version of Discounted Cumulative Gain (DCG) (Equation 3.6), to avoid that different rank sizes affect the comparisons (Equation 3.7). DCG is measured for the query q analyzing its rank up to the position k. In Equation 3.6, $rel(q, i)$ measures the relevance of the i-th element for q. In Equation 3.7, *Ideal Discounted Cumulative Gain* (IDCG) is the maximum possible DCG for query q comprising all possible results to it, which is theoretically obtained if we sort its rank placing the most relevant results first.

$$DCG(q,k) = \sum_{i=1}^{k} \frac{2^{rel(q,i)} - 1}{log_2(i+1)} \tag{3.6}$$

$$NDCG(q,k) = \frac{DCG(q,k)}{IDCG(q)} \tag{3.7}$$

For each dataset, we evaluate the effectiveness of the individual rankers, and also their correlation. Both the effectiveness and the correlation scores are used to guide the choice of base rankers. We evaluate three approaches for selecting rankers: all rankers available for each dataset; the pair composed of the two best rankers in terms of effectiveness; and the pair of rankers that present the best balance between high effectiveness and low correlation.

The second and third approaches may lead to the use of the same pair of rankers. Therefore, in cases where this happens, we also present the aggregation using the three most effective rankers. For the third approach, we select the pair of rankers R_x and R_y that maximizes the selection measure $M(R_x, R_y)$ expressed in Equation 3.8:

$$M(R_x, R_y) = \frac{1 + ef_{R_x} \times ef_{R_y}}{1 + cor(R_x, R_y)} \tag{3.8}$$

where ef_{R_x} denotes the effectiveness value for the ranker R_x, regardless the evaluation metric used (NDCG@10 or N-S Score), and $cor(R_x, R_y)$ is the correlation between R_x and R_y. This is a modified measure adapted from the one proposed in [129].

Let τ_A and τ_B be two ranks, and n be the size of these ranks. The correlation between two rankers is given by the mean correlation of their ranks with respect to each query. We adopt Jaccard's correlation, given by Equation 3.9. Other metrics were considered,

but Jaccard was the one that achieved ranker combinations for rank aggregation with the best results, in preliminary analysis that we performed, considering the possibilities for computing $cor(R_x, R_y)$ from Equation 3.8 as with Jaccard, Kendall Tau or Spearman. An equivalent conclusion was observed in [129], that investigated possibilities for ranker selection.

Kendall Tau relies on the number of discordant pairs between τ_A and τ_B. Given two response items (s_i, s_j), this pair is named discordant for τ_A and τ_B, if $\rho_{\tau_A}(s_i) > \rho_{\tau_B}(s_j)$ and $\rho_{\tau_A}(s_j) > \rho_{\tau_B}(s_i)$. Kendall Tau's correlation is given by Equation 3.10, where K_d is the number of discordant pairs and $n_d = \frac{n \times (n-1)}{2}$. Spearman correlation relies on the position disparity of each response item in the two ranks, and it is given by Equation 3.11.

$$J(\tau_A, \tau_B) = \frac{\mid \tau_A \cap \tau_B \mid}{\mid \tau_A \cup \tau_B \mid} \tag{3.9}$$

$$K_s(\tau_A, \tau_B) = 1 - \frac{K_d(\tau_A, \tau_B)}{n_d} \tag{3.10}$$

$$S(\tau_A, \tau_B) = 1 - \frac{\sum_{s_i \in \mathcal{T}_A} \mid \rho_{\mathcal{T}_A}(s_i) - \rho_{\mathcal{T}_B}(s_i) \mid}{n \times (n+1)} \tag{3.11}$$

Several state-of-the-art rank aggregation baselines are tested, along with our method, for the same candidate set of rankers: QueryRankFusion [147], RecKNNGraphCCs [107], RkGraph [106], CorGraph [105], MRA [49], RRF [32], CombSUM [52], CombMIN, CombMAX, CombMED, CombANZ, CombMNZ, BordaCount [17], Condorcet, Kemeny, and RLSim [104]. These baselines were detailed in Section 2. They are unsupervised rank aggregation methods, as it is our method, and they cover most state-of-the-art graph-based approaches, as well as some classic but still competitive ones. Because we propose an unsupervised method, we adopted unsupervised baselines to make fair comparisons. For UKBench, we also compare the results with the ones associated with the methods described in the following recent works: Bai and Bai [11], Xie et al. [140], Zheng et al. [151], Zheng et al. [150], Wang et al. [134], and Qin et al. [110].

We conduct statistical tests, using per-query paired t-test at 99% confidence level. We denote the statistical analysis with the following symbols: ▲ indicates that our method was statistically better than the baseline, ▼ means the opposite, and • means a statistical tie.

As we analyze a large number of datasets, fusion configurations (which rankers to fuse) and baselines, besides the statistical comparisons we also present the *winning number* [124] of each rank aggregation function, aiming at providing a global performance indicator per method. The winning number of a method m, W_m, regarding a performance measure P, is adapted to our context as in Equation 3.12, where D is the set of datasets, C_d is the set of our 3 pre-defined configurations for dataset d with respect to which rankers to fuse, $P_m(d, c)$ is the performance of the method m on dataset d and configuration $c \in C_d$, M is set of rank aggregation methods, and $\mathbf{1}_{P_m(d) > P_k(d)}$ is the indicator function given by Equation 3.13.

$$W_m = \sum_{d \in D} \sum_{c \in C_d} \sum_{i \in M} 1_{P_m(d,c) > P_i(d,c)} \tag{3.12}$$

Table 3.3: Results for individual rankers on textual, image, and hybrid datasets.

(a) Brodatz

Ranker	NDCG@10
LAS	0.850533
CCOM	0.726186
LBP	0.652759

(b) UW dataset

Ranker	NDCG@10
JAC	0.810729
BIC	0.746454
DICE	0.722831
BOW	0.720781
OKAPI	0.716035
JACCARD	0.701651
TF-IDF	0.658880
GCH	0.630315
COSINE	0.554767
LAS	0.514314
HTD	0.495002
QCCH	0.414249

(c) MPEG-7

Ranker	NDCG@10
ASC	0.941585
AIR	0.939424
CFD	0.930685
IDSC	0.922828
BAS	0.866098
SS	0.611481

(d) Ohsumed

Ranker	NDCG@10
BoW-cosine	0.669701
2grams-cosine	0.664120
GNF-WGU	0.662668
GNF-MCS	0.655420
2grams-Jaccard	0.651320
BoW-Jaccard	0.645711
WMD	0.427361

(e) UKBench

Ranker	N-S Score
VOC	3.54
ACC	3.37
CNN-Caffe	3.31
SCD	3.15
JCD	2.79
FCTH	2.73
CEDD	2.61

(f) Soccer

Ranker	NDCG@10
BIC	0.614818
ACC	0.592699
GCH	0.536412

Table 3.4: Correlation of individual ranks on Brodatz.

	CCOM	LAS	LBP
CCOM	1.00	0.38	0.25
LAS	0.38	1.00	0.30
LBP	0.25	0.30	1.00

$$\mathbf{1}_{P_m(d,c)>P_i(d,c)} = \begin{cases} 1 & \text{if } P_m(d,c) > P_i(d,c), \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

3.3.3 Ranker Effectiveness and Correlations

Tables 3.3a, 3.3b, 3.3c, 3.3d, 3.3e, and 3.3f report the results obtained by the individual rankers, respectively for the datasets Brodatz, UW, MPEG-7, Ohsumed, UKBench, and Soccer. The rankers are presented sorted by their results. It can be noticed large variability in rankers' results. Furthermore, rankers perform differently depending on the dataset, possibly providing complementary views. For example, JACCARD was better than COSINE in UW, but the opposite happened for the Ohsumed dataset.

Tables 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9 report the Jaccard's correlations between ranks for the individual rankers used, respectively for the datasets Brodatz, UW, MPEG-7, Ohsumed, UKBench, and Soccer. These correlations, along with the individual rankers' effectiveness, provide useful insights with respect to which rankers should be combined. In Ohsumed, WMD shows very low correlation to the other rankers, even though it was the worst effective ranker.

Table 3.5: Correlation of individual ranks on UW.

	BIC	GCH	HTD	JAC	LAS	QCCH	BOW	COSINE	DICE	JACCARD	OKAPI	TF-IDF
BIC	1.00	0.29	0.12	0.27	0.12	0.11	0.14	0.11	0.14	0.13	0.13	0.11
GCH	0.29	1.00	0.11	0.18	0.11	0.10	0.11	0.08	0.11	0.10	0.09	0.08
HTD	0.12	0.11	1.00	0.12	0.12	0.12	0.09	0.07	0.09	0.08	0.07	0.07
JAC	0.27	0.18	0.12	1.00	0.11	0.10	0.14	0.11	0.15	0.14	0.13	0.12
LAS	0.12	0.11	0.12	0.11	1.00	0.16	0.08	0.06	0.08	0.08	0.07	0.07
QCCH	0.11	0.10	0.12	0.10	0.16	1.00	0.08	0.06	0.08	0.07	0.06	0.06
BOW	0.14	0.11	0.09	0.14	0.08	0.08	1.00	0.22	0.44	0.42	0.30	0.25
COSINE	0.11	0.08	0.07	0.11	0.06	0.06	0.22	1.00	0.32	0.32	0.36	0.45
DICE	0.14	0.11	0.09	0.15	0.08	0.08	0.44	0.32	1.00	0.85	0.37	0.37
JACCARD	0.13	0.10	0.08	0.14	0.08	0.07	0.42	0.32	0.85	1.00	0.38	0.37
OKAPI	0.13	0.09	0.07	0.13	0.07	0.06	0.30	0.36	0.37	0.38	1.00	0.60
TF-IDF	0.11	0.08	0.07	0.12	0.07	0.06	0.25	0.45	0.37	0.37	0.60	1.00

Table 3.6: Correlation of individual ranks on MPEG-7.

	AIR	ASC	BAS	CFD	IDSC	SS
AIR	1.00	0.31	0.27	0.30	0.30	0.18
ASC	0.31	1.00	0.33	0.37	0.70	0.20
BAS	0.27	0.33	1.00	0.48	0.32	0.28
CFD	0.30	0.37	0.48	1.00	0.36	0.26
IDSC	0.30	0.70	0.32	0.36	1.00	0.19
SS	0.18	0.20	0.28	0.26	0.19	1.00

Table 3.7: Correlation of individual ranks on Ohsumed.

	BoW-cosine	BoW-Jaccard	2grams-cosine	2grams-Jaccard	GNF-MCS	GNF-WGU	WMD
BoW-cosine	1.00	0.56	0.48	0.45	0.49	0.55	0.10
BoW-Jaccard	0.56	1.00	0.41	0.50	0.51	0.54	0.11
2grams-cosine	0.48	0.41	1.00	0.64	0.51	0.58	0.10
2grams-Jaccard	0.45	0.50	0.64	1.00	0.55	0.61	0.10
GNF-MCS	0.49	0.51	0.51	0.55	1.00	0.73	0.10
GNF-WGU	0.55	0.54	0.58	0.61	0.73	1.00	0.10
WMD	0.10	0.11	0.10	0.10	0.10	0.10	1.00

Table 3.8: Correlation of individual ranks on UKBench.

	ACC	VOC	CNN-Caffe	SCD	JCD	FCTH	CEDD
ACC	1.00	0.23	0.22	0.31	0.23	0.22	0.21
VOC	0.23	1.00	0.24	0.22	0.21	0.20	0.20
CNN-Caffe	0.22	0.24	1.00	0.21	0.20	0.19	0.19
SCD	0.31	0.22	0.21	1.00	0.26	0.26	0.23
JCD	0.23	0.21	0.20	0.26	1.00	0.39	0.53
FCTH	0.22	0.20	0.19	0.26	0.39	1.00	0.28
CEDD	0.21	0.20	0.19	0.23	0.53	0.28	1.00

Table 3.9: Correlation of individual ranks on Soccer.

	BIC	ACC	GCH
BIC	1.00	0.46	0.27
ACC	0.46	1.00	0.30
GCH	0.27	0.30	1.00

3.3.4 Rank Aggregation Results

We report the rank aggregation results obtained by our method and by the baselines, for each dataset and each of the three combinations of rankers. From the evaluation procedure previously presented, the following combinations of rankers were selected per dataset:

- Brodatz: all 3 rankers; LAS + CCOM; LAS + LBP.
- Soccer: all 3; BIC + ACC; BIC + GCH.
- MPEG-7: all 6; ASC + AIR; AIR + CFD.
- Ohsumed: all 7; BoW-cosine + 2grams-cosine; BoW-cosine + WMD.
- UKBench: all 7; VOC + ACC; VOC + ACC + CNN-Caffe.
- UW: all 12; JAC + BIC; JAC + OKAPI.

Recall that the use of LAS + LBP and BIC + GCH for the Brodatz and the Soccer datasets, respectively, were defined according to Equation 3.8. The same approach was used for the other datasets. For UKBench, both second and third selection approaches lead to the same pair of rankers, so we also present the aggregation using its three most effective rankers.

We performed experiments for different values of L in the range $\{2, 4, 6, 8, 10, 12, 14, 16, 20\}$ for different datasets. The experimental results, using the fusion of all selected rankers for each dataset is presented in Figure 3.4. As expected, the effectiveness increased when L goes from 2 to 10, and from that point on, the effectiveness measure roughly stabilized.

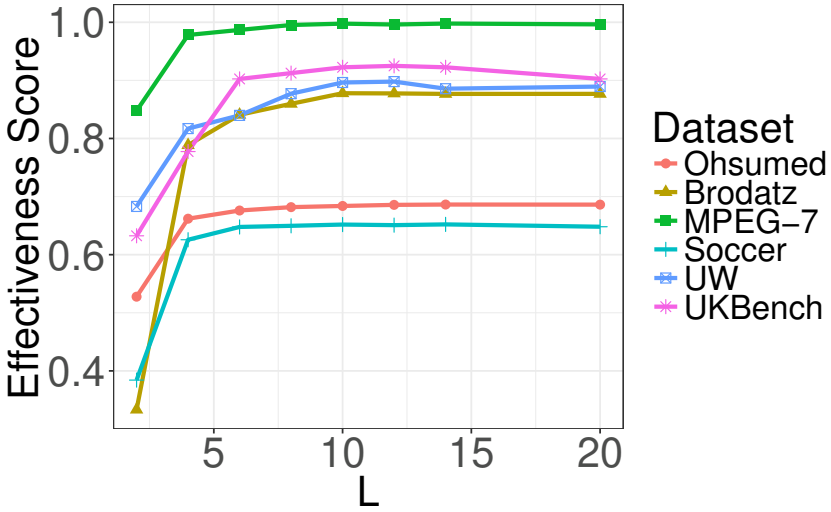


Figure 3.4: Stabilizing effect of the cut-off parameter L in the effectiveness performance. N-S scores for UKBench are rescaled to $[0, 1]$ for consistency in the plot, while the rest corresponds to NDCG@10.

In order to evaluate the impact of different fusion graph comparators in our method, we present in Table 3.10 the effectiveness scores achieved by either WGU or MCS, measured

by N-S for UKBench, and NDCG@10 for the rest. The performance is slightly better with WGU than with MCS in absolute values, and WGU was statistically superior in two out of six cases. The evaluation comprised all datasets, using the fusion of all selected rankers for them. For this reason, in the remaining experiments performed, WGU is adopted.

Table 3.10: Effect of different fusion graph comparators in the effectiveness performance.

Collection	Effectiveness		Difference (p.p.)	Statistical Difference
	WGU	MCS		
Ohsumed	0.683835	0.677880	0.5955	▲
Brodatz	0.878995	0.878675	0.0320	•
MPEG-7	0.997658	0.997821	-0.0163	•
Soccer	0.651828	0.651982	-0.0154	•
UW	0.873607	0.874543	-0.0936	•
UKBench	3.69	3.67	0.5	▲

Tables 3.11, 3.12, 3.13, 3.14, 3.15, and 3.16 report the results obtained, respectively for Ohsumed, Brodatz, MPEG-7, Soccer, UW, and UKBench.

Most baselines presented results worse than the best individual rankers, but our method overcame the individual rankers in all scenarios. It can be seen that most baselines are dramatically affected by bad individual ranks, in the sense that the addition of a poor ranker into the aggregation function leads to poor effectiveness. This may be seen as a limitation of unsupervised rank aggregation functions in general. Our method, on the contrary, was shown to be much less sensitive to this search scenario. For the Ohsumed dataset, for example, WMD performed much worse than BoW-cosine, but, as they produce low correlated ranks, their fusion still yielded a better ranker.

The criteria adopted to choose pairs of rankers for combination, based on effectiveness and correlation, led to pairs whose aggregated results surpassed pairs formed by the most effective rankers for the MPEG-7 and UW datasets. In most cases, the selection of the most effective base rankers yields suitable results.

In Ohsumed, Brodatz, and MPEG-7, the aggregation of all rankers performed better than the combination of selected pairs of rankers. These results demonstrate that even less competitive rankers can contribute to improving retrieval tasks when used in the aggregation.

While the ranker selection criteria of using all rankers led to top performance in half the datasets, it also demands additional processing cost. The analysis of the three ranker selection approaches allows us to conclude that the ranker selection of the two most competitive rankers per dataset is an overall good choice, but subjected to improvement after a careful empirical evaluation of the other approaches in the desired scenario.

We summarize in Table 3.17 the results achieved by the rankers in each dataset, and report our gains over them, in percentage gain. *FG* was able to present significant gains over the rankers.

Our method achieved either top or very competitive performance in all datasets and combinations of rankers tested. For 6 datasets with 3 configurations each, and for 16 state-of-the-art baselines, *FG* was statistically worse only in 7 out of 288 comparisons.

Table 3.11: Results for rank aggregation on Ohsumed.

Method	NDCG@10		
	BoW-cosine + BoW-Jaccard + 2grams-cosine + 2grams-Jaccard + GNF-MCS + GNF-WGU + WMD	BoW-cosine + 2grams-cosine	BoW-cosine + WMD
FG	0.683835	0.683472	0.676760
RecKNNGraphCCs	▲ 0.676234	▲ 0.679728	▲ 0.667750
CombSUM	▲ 0.666997	▲ 0.671868	▲ 0.598441
CombMED	▲ 0.666997	▲ 0.671868	▲ 0.598441
CombMNZ	▲ 0.666929	▲ 0.671869	▲ 0.598261
QueryRankFusion	▲ 0.651279	▲ 0.671258	▲ 0.669704
MRA	▲ 0.666045	▲ 0.670357	▲ 0.582049
RRF	▲ 0.665793	▲ 0.671294	▲ 0.571016
BordaCount	▲ 0.660197	▲ 0.671147	▲ 0.570466
Condorcet	▲ 0.619869	▲ 0.670235	▲ 0.569906
CombMAX	▲ 0.611777	▲ 0.671305	▲ 0.597080
CombANZ	▲ 0.567671	▲ 0.670550	▲ 0.595983
Kemeny	▲ 0.543564	▲ 0.665817	▲ 0.526588
CombMIN	▲ 0.502482	▲ 0.666559	▲ 0.591361
RLSim	▲ 0.434614	▲ 0.639004	▲ 0.579972
CorGraph	▲ 0.487177	▲ 0.497431	▲ 0.456434
RkGraph	▲ 0.289045	▼ 0.688443	▲ 0.288436

Table 3.12: Results for rank aggregation on Brodatz.

Method	NDCG@10		
	LAS+CCOM+LBP	LAS+CCOM	LAS+LBP
RecKNNGraphCCs	• 0.877882	▼ 0.882903	▼ 0.839717
FG	0.878995	0.872084	0.835624
RkGraph	▲ 0.812659	▲ 0.861250	▲ 0.788682
QueryRankFusion	▲ 0.850263	▲ 0.850438	▲ 0.808562
CombMNZ	▲ 0.822887	▲ 0.827517	▲ 0.787922
CombSUM	▲ 0.812971	▲ 0.826075	▲ 0.784971
CombMED	▲ 0.812971	▲ 0.826075	▲ 0.784971
CombMAX	▲ 0.787828	▲ 0.818125	▲ 0.776842
RRF	▲ 0.818656	▲ 0.817139	▲ 0.788840
BordaCount	▲ 0.805699	▲ 0.814664	▲ 0.785836
MRA	▲ 0.822778	▲ 0.813396	▲ 0.788883
CombANZ	▲ 0.763987	▲ 0.812431	▲ 0.769743
Condorcet	▲ 0.781129	▲ 0.809929	▲ 0.781781
CorGraph	▲ 0.749420	▼ 0.895623	▲ 0.719204
CombMIN	▲ 0.713228	▲ 0.794631	▲ 0.752268
Kemeny	▲ 0.719680	▲ 0.786537	▲ 0.757349
RLSim	▲ 0.633157	▲ 0.756053	▲ 0.724879

Table 3.13: Results for rank aggregation on MPEG-7.

Method	NDCG@10		
	AIR + CFD + ASC + IDSC + BAS + SS	ASC + AIR	AIR + CFD
FG	0.997658	0.994729	0.995886
RecKNNGraphCCs	• 0.998052	• 0.995160	• 0.997267
RkGraph	▲ 0.826119	▼ 0.999350	▲ 0.992078
CorGraph	▲ 0.992456	▲ 0.962951	▲ 0.961460
RRF	▲ 0.980638	▲ 0.957684	▲ 0.954499
MRA	▲ 0.980086	▲ 0.950442	▲ 0.946144
CombMNZ	▲ 0.976832	▲ 0.942705	▲ 0.932234
BordaCount	▲ 0.974697	▲ 0.954296	▲ 0.951316
CombSUM	▲ 0.969212	▲ 0.941585	▲ 0.930685
CombMED	▲ 0.969212	▲ 0.941585	▲ 0.930685
QueryRankFusion	▲ 0.940976	▲ 0.941762	▲ 0.941271
CombMAX	▲ 0.930012	▲ 0.941585	▲ 0.930685
Condorcet	▲ 0.911624	▲ 0.950122	▲ 0.947416
CombANZ	▲ 0.862366	▲ 0.938649	▲ 0.927035
Kemeny	▲ 0.792694	▲ 0.940479	▲ 0.929906
CombMIN	▲ 0.626645	▲ 0.902798	▲ 0.888723
RLSim	▲ 0.444817	▲ 0.902798	▲ 0.888723

Table 3.14: Results for rank aggregation on Soccer.

Method	NDCG@10		
	BIC+ACC+GCH	BIC+ACC	BIC+GCH
FG	0.651828	0.655332	0.622217
RkGraph	• 0.653623	• 0.656422	• 0.628563
CorGraph	▲ 0.645004	▲ 0.643505	• 0.623627
RecKNNGraphCCs	▲ 0.637537	▲ 0.640729	• 0.618704
QueryRankFusion	▲ 0.613732	▲ 0.613659	▲ 0.598862
BordaCount	▲ 0.603156	▲ 0.613205	▲ 0.589633
RRF	▲ 0.604119	▲ 0.613005	▲ 0.590819
CombSUM	▲ 0.604575	▲ 0.611546	▲ 0.588667
CombMED	▲ 0.604565	▲ 0.611546	▲ 0.588667
CombMNZ	▲ 0.605567	▲ 0.611269	▲ 0.589202
MRA	▲ 0.605971	▲ 0.611017	▲ 0.588399
CombANZ	▲ 0.587048	▲ 0.610981	▲ 0.582010
Condorcet	▲ 0.593809	▲ 0.611049	▲ 0.589266
CombMAX	▲ 0.591911	▲ 0.609345	▲ 0.584983
Kemeny	▲ 0.578919	▲ 0.607451	▲ 0.578043
CombMIN	▲ 0.570258	▲ 0.606144	▲ 0.576877
RLSim	▲ 0.506736	▲ 0.570744	▲ 0.545591

Table 3.15: Results for rank aggregation on UW.

Method	NDCG@10		
	JAC + BIC + DICE + BOW + OKAPI + JACCARD + TF-IDF + GCH + COSINE + LAS + HTD + QCCH	JAC + BIC	JAC + OKAPI
CorGraph	▼ 0.896341	▲ 0.842665	▼ 0.933452
FG	0.873607	0.854473	0.882776
RecKNNGraphCCs	▲ 0.869448	▲ 0.843423	• 0.882035
RkGraph	▲ 0.746804	▲ 0.841127	▲ 0.866544
MRA	▲ 0.815983	▲ 0.797292	▲ 0.786995
RRF	▲ 0.815779	▲ 0.798502	▲ 0.795143
CombMNZ	▲ 0.806416	▲ 0.793488	▲ 0.814850
BordaCount	▲ 0.789620	▲ 0.797677	▲ 0.788127
CombSUM	▲ 0.769227	▲ 0.793057	▲ 0.812383
CombMED	▲ 0.769227	▲ 0.793057	▲ 0.812383
QueryRankFusion	▲ 0.747281	▲ 0.792681	▲ 0.807250
Condorcet	▲ 0.743168	▲ 0.795304	▲ 0.780080
CombMAX	▲ 0.691427	▲ 0.786912	▲ 0.802788
CombANZ	▲ 0.596119	▲ 0.784183	▲ 0.795284
Kemeny	▲ 0.471099	▲ 0.773449	▲ 0.739709
CombMIN	▲ 0.359668	▲ 0.773776	▲ 0.769389
RLSim	▲ 0.330593	▲ 0.740222	▲ 0.768275

Table 3.16: Results for rank aggregation on UKBench.

Method	N-S Score		
	VOC + ACC + CNN-Caffe + SCD + JCD + FCTH + CEDD	VOC + ACC	VOC + ACC + CNN-Caffe
FG	3.69	3.83	3.90
RecKNNGraphCCs	▲ 3.67	▲ 3.81	▲ 3.87
QueryRankFusion	▲ 3.60	▲ 3.78	▲ 3.86
MRA	▲ 3.52	▲ 3.50	▲ 3.77
CombSUM	▲ 3.55	▲ 3.60	▲ 3.76
CombMED	▲ 3.55	▲ 3.60	▲ 3.76
CombMNZ	▲ 3.53	▲ 3.60	▲ 3.76
BordaCount	▲ 3.55	▲ 3.60	▲ 3.76
RRF	▲ 3.52	▲ 3.60	▲ 3.76
Condorcet	▲ 3.64	▲ 3.58	▲ 3.75
CombMAX	▲ 3.13	▲ 3.52	▲ 3.48
RkGraph	▲ 3.03	▲ 3.50	▲ 3.54
CombANZ	▲ 2.83	▲ 3.42	▲ 3.28
Kemeny	▲ 2.51	▲ 3.37	▲ 3.14
CombMIN	▲ 2.35	▲ 3.36	▲ 3.09
CorGraph	▲ 2.44	▲ 2.91	▲ 2.77
RLSim	▲ 1.09	▲ 2.73	▲ 1.89

Table 3.17: Effectiveness of rankers compared to our method, in textual, image, and hybrid datasets.

(a) Brodatz

Method	NDCG@10	Gains (%)
FG	0.878995	
LAS	0.850533	3.35
CCOM	0.726186	21.04
LBP	0.652759	33.66

(b) UW dataset

Method	NDCG@10	Gains (%)
FG	0.873607	
JAC	0.810729	7.76
BIC	0.746454	17.03
DICE	0.722831	20.86
BOW	0.720781	21.20
OKAPI	0.716035	22.01
JACCARD	0.701651	24.51
TF-IDF	0.658880	32.59
GCH	0.630315	38.60
COSINE	0.554767	57.47
LAS	0.514314	69.86
HTD	0.495002	76.49
QCCH	0.414249	110.89

(c) MPEG-7

Method	NDCG@10	Gains (%)
FG	0.997658	
ASC	0.941585	5.96
AIR	0.939424	6.20
CFD	0.930685	7.20
IDSC	0.922828	8.11
BAS	0.866098	15.19
SS	0.611481	63.15

(d) Ohsumed

Method	NDCG@10	Gains (%)
FG	0.683835	
BoW-cosine	0.669701	2.11
2grams-cosine	0.664120	2.97
GNF-WGU	0.662668	3.19
GNF-MCS	0.655420	4.34
2grams-Jaccard	0.651320	4.99
BoW-Jaccard	0.645711	5.90
WMD	0.427361	60.01

(e) UKBench

Method	N-S Score	Gains (%)
FG	3.90	
VOC	3.54	10.17
ACC	3.37	15.73
CNN-Caffe	3.31	17.83
SCD	3.15	23.81
JCD	2.79	39.79
FCTH	2.73	42.86
CEDD	2.61	49.43

(f) Soccer

Method	NDCG@10	Gains (%)
FG	0.655332	
BIC	0.614818	6.59
ACC	0.592699	10.57
GCH	0.536412	22.17

Besides, it was top-1 in 4 out of 6 datasets (Ohsumed, MPEG-7, Soccer, and UKBench), and top-2 in Brodatz and UW.

We present in Figure 3.5 the winning numbers achieved per rank aggregation function, as an alternative way to contrast them globally. *FG* was broadly superior to the majority of baselines, according to our experimental evaluation comprising 3 aggregation approaches for each of the 6 distinct datasets used.

Table 3.18 presents the results of our ranker for UKBench, together with seven additional baselines. The table reports the results from Table 3.16, obtained using ACC + VOC + CNN-Caffe, together with results reported by the other baselines. QueryRankFusion is presented twice, one regarding their own reported result, and another considering the same input rankers as ours. It is worth to notice that three of these additional baselines performed worse than some classic and much simpler rank aggregation functions. Again, our method achieved the best performance.

Our rank fusion has been shown to be effective in combining contextual information from different ranks, along with the intrinsic relationships that the retrieved objects have to each other in their own ranks. Also, our procedure to rank objects based on fusion graphs considers these fusions automatically, relying on such graphs without any other intermediate steps, such as training or parameter tuning.

3.3.5 Efficiency Analysis

Section 3.2.3 presents the asymptotic cost of our method. The time for performing a query is around the sum of the slowest time to produce an isolated rank plus the time to produce the final rank based on fusion graphs. Table 3.19 presents, per dataset, the mean time (in milliseconds) spent per query, and the mean offline time (in seconds) to produce the fusion graphs from the response set. For both values, we report the mean

Table 3.18: State-of-the-art results on UKBench. Results marked with * were obtained using ACC + VOC + CNN-Caffe.

Method	N-S Score
FG	**3.90***
Xie et al. [140]	3.89
RecKNNGraphCCs	3.87*
Bai and Bai [11]	3.86
QueryRankFusion	3.86*
Zheng et al. [151]	3.84
QueryRankFusion	3.83
MRA	3.77*
CombSUM	3.76*
CombMED	3.76*
CombMNZ	3.76*
BordaCount	3.76*
RRF	3.76*
Condorcet	3.75*
Wang et al. [134]	3.68
Qin et al. [110]	3.67
Zheng et al. [150]	3.57
RkGraph	3.54*
CombMAX	3.48*
CombANZ	3.28*
Kemeny	3.14*
CombMIN	3.09*
CorGraph	2.91*
RLSim	1.89*

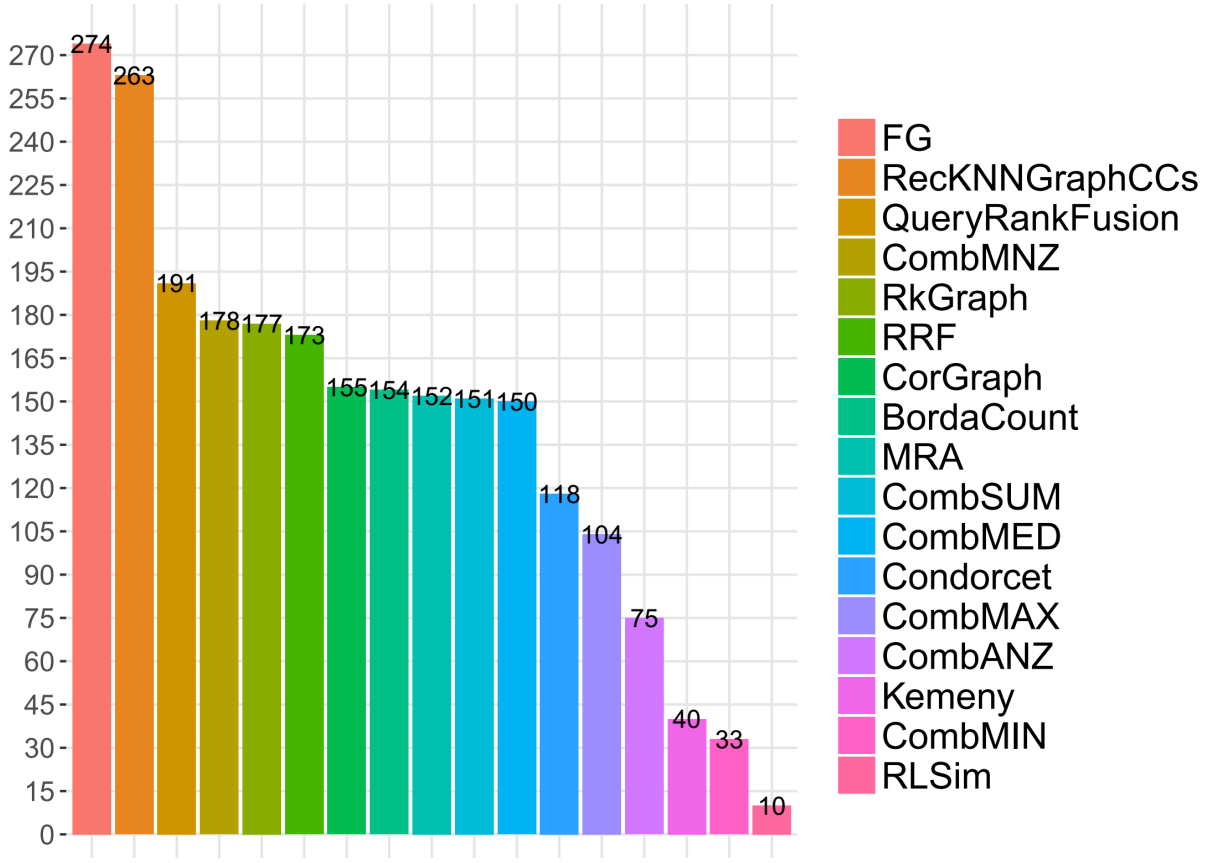


Figure 3.5: Winning numbers achieved per rank aggregation function.

time of 5 independent measurements, taken on an Intel Core i7-7500U CPU @ 2.70GHz with 16GB of RAM. For all datasets, the search time was reasonable, given the high gains in effectiveness provided by our method. The offline time is also low, due to unsupervised nature of our method.

Table 3.19: Rank aggregation time per query, and offline time.

Dataset	Rank Aggregation Time (in ms)	Offline Time (in sec)
Brodatz	8.76 ± 2.70	2.28 ± 0.75
MPEG-7	25.73 ± 2.76	5.92 ± 0.60
Ohsumed	101.13 ± 16.88	38.12 ± 3.78
Soccer	4.60 ± 2.58	0.63 ± 0.16
UW	29.29 ± 7.80	10.92 ± 1.48
UKBench	21.30 ± 6.68	25.34 ± 2.02

3.4 Conclusions

Distinct feature extractors – and by extension different retrieval models as well – provide different and complementary views of textual and multimedia documents in retrieval tasks. Therefore, combining their capabilities for a more effective retrieval without the need of user intervention remains a relevant yet challenging task.

In this work, a novel unsupervised graph-based rank aggregation method was proposed. Our approach models the rank fusion task by means of a fusion graph and derives a novel retrieval score, directly based on the graph structure. The method was extensively evaluated on textual, images, and hybrid datasets comprising ad-hoc retrieval tasks, achieving superior effectiveness scores than the best isolated features and several baselines.

We started a new retrieval paradigm by means of graph-based rank fusions. From that, alternative graph-based formulations, or retrieval scores, can be explored. For example, one may be interested in fusion graphs that take both ranks and raw features into account. Or maybe one may be interested in a retrieval score that analyzes retrieval effectiveness along with diversification. New research venues can arise based on our proposal.

A practical implication of our solution is that scenarios involving changes in the response set or the rankers should trigger an update in the offline stage, due to the fact that response fusion graphs are previously computed. Although the cost regarding the offline stage is reasonable, as discussed in Section 3.2.3, frequent changes would in theory lead to the need for frequent updates. We could mitigate this issue by making partial or approximate updates, such as generating fusion graphs only for new objects, or even postpone updates until necessary. We leave the investigation of this practical aspect for future work.

As a future work, we intend to evaluate our method against supervised [71, 97] or semi-supervised [38, 109] techniques. Finally, we want to explore rank-fusion vector representations based on graphs. The goal is to take advantage of existing solutions (e.g., indexing schemes) to make our fusion method even more scalable.

Chapter 4

Fusion Vectors: Embedding Graph Fusions for Efficient Unsupervised Rank Aggregation

4.1 Introduction

Although recent rank aggregation functions are promising with respect to effectiveness, such as graph-based approaches [107, 147], many of them do not handle multimodality or have not been validated in such scenarios [11, 82, 107, 140]. Yet, very few works have already investigated the proposal or representation models in the context of rank aggregation functions [11]. Besides, even recent proposals are not strictly bundled with efficiency [107]. Nevertheless, information retrieval typically has to deal with large datasets, thus demanding efficient retrieval. On the other hand, a number of works from related research fields have been proposed regarding indexing structures, embedding formulations [23, 44, 116], and approximate search [89]. Here we investigate the applicability of such initiatives in the context of rank aggregation, while targeting multimodality and representation models.

We present a rank aggregation formulation, derived from the previous proposal, which is based on the embedding of Fusion Graphs as vectors, called Fusion Vectors. This alternative formulation, although theoretically subject to some loss of information contained in graphs, has some advantages:

- greater availability of algorithms and techniques when we compare vector domains and graph domains;
- improved storage and compression capability;
- possibility of using indexing techniques.

To our knowledge, not only the elaboration of individual rank fusion graphs is innovative, but also the proposal of embedding these types of graph representations. This work is one of the first to present and evaluate this approach.

We also propose an indexing mechanism for fusion vectors, which stores the resulting vectors from the collection, in order to provide time-efficient query resolution. This also

expands the applicability of the method to large-scale scenarios, thus solving a typical limitation of various rank aggregation related works, particularly those based on graphs. The solution is overall unsupervised, so that no labeled data are required.

The contributions of the chapter are:

1. The proposal of an innovative rank aggregation function, that it is unsupervised, intrinsically multimodal, and targeted for fast retrieval and top effectiveness performance;

2. The introduction of embedding approaches for graph-based rank-aggregation elements;

3. The proposal of an strategy for indexing and approximate retrieval based on rank-fusion vectors.

4.2 Fast Rank Aggregation Retrieval

An overview of our rank aggregation proposal is shown in Figure 4.1, which highlights two stages – offline and online – and four enumerated components. The offline stage comprehends the modeling of the response set in terms of multiple rankers, and its indexing for further retrieval; while the online stage refers to the steps employed in a search session. The solution is composed of four main generic components, briefly described here and detailed in the following sections. The first two components are used in both stages.

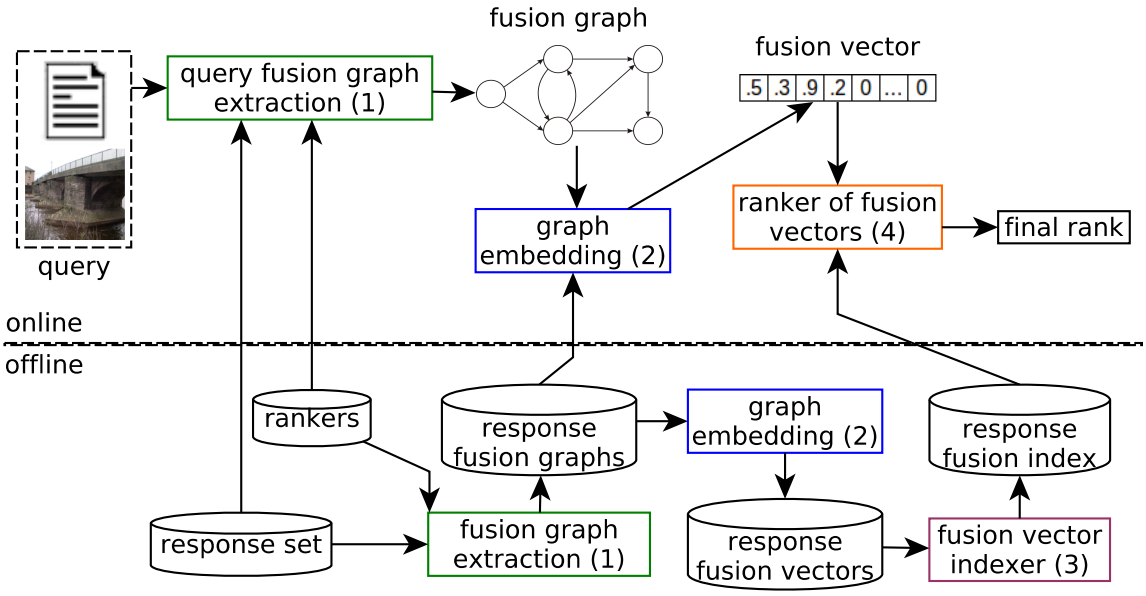


Figure 4.1: Schematic view of the proposed method.

The *fusion graph extraction* component (1) generates a fusion graph for a given query sample. A fusion graph $\mathcal{G}$ consists of an aggregated representation of multiple ranks for a query, thus capturing and correlating information of its associated multiple ranks. This formulation is presented in Section 4.2.1.

Graph Embedding (2) projects fusion graphs into a vector space model, producing a corresponding fusion vector $\mathcal{V}$ per fusion graph. We propose and discuss some possible embedding formulations in Section 4.2.2.

Fusion vector indexer (3) generates an index of fusion vectors, from which it is possible to retrieve relevant items for a given query, as long as both the query and the response items are previously represented as fusion vectors. By means of the response fusion index, efficient searches of multi-ranked objects can be performed. Although the ranks could be generated from the response fusion vectors directly, through brute-force search, the indexing step is important to promote sub-linear query processing time.

At the end, a *ranker of fusion vectors* (4) produces a rank of objects for a certain query object, according to the similarity of their respective fusion vectors. These last two components, *fusion vector indexer* and *ranker of fusion vectors*, are detailed in Section 4.2.3.

4.2.1 Fusion Graph Extraction

This component produces a fusion graph $\mathcal{G}$ for a given query sample q based on its ranks. A fusion graph is a graph-based encoding of multiple ranks for q, that intrinsically encapsulates and correlates ranks. We follow the fusion graph formulation from Section 3.2.1, which defined a procedure to extract a fusion graph $\mathcal{G}$, for q, based on its ranks and ranks' inter-relationships. We summarize the approach here. For a more detailed explanation, its reasoning and discussion, please refer to Section 3.2.1.

It was also defined a retrieval model based on fusion graphs, hereby referred to as *FG*. We adopt *FG* as a baseline in this chapter. Other graph-based rank aggregations than ours could also be explored, but we leave this study for future work.

Let $\mathcal{T}_q$ be a set of m ranks for the query q, with sizes up to a certain limit L, and obtained which respect to m rankers, over a dataset S of size n. Besides, consider that the ranks from every response item $s_i \in S$, regarding the m rankers, are pre-computed in offline stage. Also, let $\varsigma_{\tau_q}(s_i, s_j)$ be the similarity score between s_i and s_j with respect to the same descriptor $\mathcal{D}$ and comparator $\mathcal{C}$ from the ranker $R(\mathcal{D}, \mathcal{C})$ that produced τ_q for q.

The fusion graph extraction is a mapping function $\mathcal{T}_q \mapsto \mathcal{G}$, and works in $O(m^2L^2)$. $\mathcal{G}$, for an object q, includes all response items from each rank $\tau_q \in \mathcal{T}_q$, as vertices. Vertices are connected by taking into account the degree of relationship between their corresponding response items, and the degree of their relationships to q. The weight of a vertex v_A, expressed by $w(v_A)$, is given by Equation 3.2. The weight of an edge $e_{A,B}$, expressed by $w(e_{A,B})$, is given by Equation 3.3.

4.2.2 Fusion Graph Embedding

Let $\mathbb{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n\}$ be the fusion graph set related to the response set of a certain collection. From $\mathbb{G}$, a fusion graph embedding function $\mathcal{E}$ defines a vector space in order to project a fusion graph $\mathcal{G}(V, E)$ into that space as a fusion vector $\mathcal{V}$, i.e., $\mathcal{V} = \mathcal{E}(\mathcal{G})$ for any $\mathcal{G}$.

A fusion vector is a representation of multi-ranked objects, and allows efficient storage and search, as vectors are commonly much easier to manage than graphs. Dissimilarity

scores between fusion vectors can be obtained by traditional vector comparators, such as Jaccard, cosine, or Euclidean functions. Based on fusion vectors, a retrieval system of multi-ranked objects can be further established.

$\mathcal{E}$ can be defined by unsupervised or supervised approaches. We focus on unsupervised approaches. We propose and evaluate three possible formulations, each one targeting different embedding categories [23]: vertex-based, hybrid, and kernel-based. Let $w_{\mathcal{G}}(v)$ be the weight of the vertex v, if $v \in \mathcal{G}$, otherwise 0. Similarly, let $w_{\mathcal{G}}(e)$ be the weight of the edge e, if $e \in \mathcal{G}$, otherwise 0. Also, let d be the dimensionality of the vector space model defined by $\mathcal{E}$, such that $\mathcal{V} \in \mathbb{R}^{\mathrm{d}}$, and $n = |\mathbb{G}|$.

Vertex-based Embedding

$\mathcal{E}_V$ is the first and simplest of our proposed embedding formulations, which derives $\mathcal{V}$ from the vertices of $\mathcal{G}$. For $\mathcal{E}_V$, there is one vector attribute relative to each response object, therefore $d = n$, and a fusion vector is defined as

$$\mathcal{V} = (u_1, \ldots, u_i, \ldots, u_d), \tag{4.1}$$

where $1 \leq i \leq d$, $u_i = w_{\mathcal{G}}(v_i)$.

Despite that vector space increases linearly to the collection size, the resulting fusion vectors are mainly sparse, i.e., composed of few non-zero entries, which allows this embedding formulation to be efficient for storage and for dissimilarity comparisons.

Hybrid Embedding

$\mathcal{E}_H$ is an embedding formulation that, different from $\mathcal{E}_V$, derives the fusion vector from both the vertices and edges of $\mathcal{G}$, therefore called a *hybrid* embedding. In $\mathcal{E}_H$, each response object contributes to one attribute in the vector space. Besides, each possible edge linking two distinct vertices, $e(v_i, v_j)$, contributes to an additional vector attribute, but we handle inverted pairs – $e(v_i, v_j)$ and $e(v_j, v_i)$ – to refer to the same attribute, as if the edges were undirected. Hence, the vector space has dimensionality

$$d = n + (\frac{n^2}{2} - n) = \frac{n^2}{2}. \tag{4.2}$$

The fusion vector is defined as

$$\mathcal{V} = (u_1, \ldots, u_i, \ldots, u_n, x_1, \ldots, x_k, \ldots, x_m), \tag{4.3}$$

where $1 \leq i \leq n$, $1 \leq k \leq m$, $m = \frac{n^2}{2} - n$, $u_i = w_{\mathcal{G}}(v_i)$, $i < j$, and $x_k = w_{\mathcal{G}}(e_{v_i,v_j}) + w_{\mathcal{G}}(e_{v_j,v_i})$.

$\mathcal{E}_H$ has the benefit over $\mathcal{E}_V$ in incorporating proximity information, at a cost of leading to a representation with more dimensions. Comparing both, $\mathcal{E}_H$ is expected to gain in effectiveness and lose in efficiency.

Kernel-based Embedding

In a kernel-based embedding, a graph is represented as a vector containing the frequencies of elementary substructures that are decomposed from that graph [23]. In this context, a graph kernel defines an atomic substructure, such as a subgraph of fixed size (graphlet), a subtree pattern, or a random walk.

$\mathcal{E}_K$ is the third proposed embedding formulation, which extends Bag of Graphs (BoG) [116] – a kernel-based embedding framework – to the rank aggregation domain. To the best of our knowledge, this is the first work that extends BoG to this scenario. BoG has been extended to the textual domain, and discussed in detail, in [44].

BoG is a general framework for graph embedding, but requires some functions to be explicitly defined according to the target scenario. The vector space is defined by a *codebook*, which is a set of attributes called *codewords*. Codewords are common local graph patterns, based on subgraphs either selected as centroids of a subgraph clustering procedure or by random selection. The schematic view is indicated in Figure 4.2, and we extend BoG in order to promote $\mathcal{E}_K$, using the following definitions:

- Graph of Interest (GoI): a pattern of $\mathcal{G}$, so that a set of valid subgraphs of $\mathcal{G}$ can be extracted. For every vertex $v \in \mathcal{G}$, we derive one undirected connected graph containing: (1) v; (2) all direct incident vertices starting from v; and (3) the edges linking them. We preserve both vertex weights and edge weights into the subgraph.

- GoI Dissimilarity Function: provides a dissimilarity score between two subgraphs. We adopt MCS (Equation 3.4), which computes the dissimilarity score based on maximum common subgraphs, and can be efficiently implemented linearly on the number of vertices [44].

- Codebook Generation: a subset of the training GoI's must be selected to compose the codebook. BoG suggests clustering or random selection. We perform a graph clustering using MeanShift [31], adapted to work with a distance matrix as input, which we compute with the GoI Dissimilarity Function.

- Assignment: defines an activation value correlating a subgraph to a vector attribute. We adopt Soft Assignment, which employs a kernel function that establishes, for an input subgraph, a score for each graph attribute [116]. Soft Assignment is given by Equation 4.4 [44], where S is the set of subgraphs for a certain input graph, a_{ij} is the assignment value between the subgraph $s_i \in S$ and the attribute w_j, d is the vocabulary size, $D(s_i, w_k)$ computes the GoI dissimilarity between s_i and w_k, and $K(x) = \frac{exp(-\frac{x^2}{2\sigma^2})}{\sigma\sqrt{2\pi}}$ is a Gaussian applied to smooth the dissimilarities [131]. σ allows the smoothness control.

- Pooling: summarizes assignments, producing the final vector (Histogram in Figure 4.2). We adopt Average Pooling, which weights the j-th vector attribute as the percentage of associations of the input sample subgraphs to the j-th graph attribute. This is given by Equation 4.5 [44], where $v[j]$ refers to the j-th vector attribute.

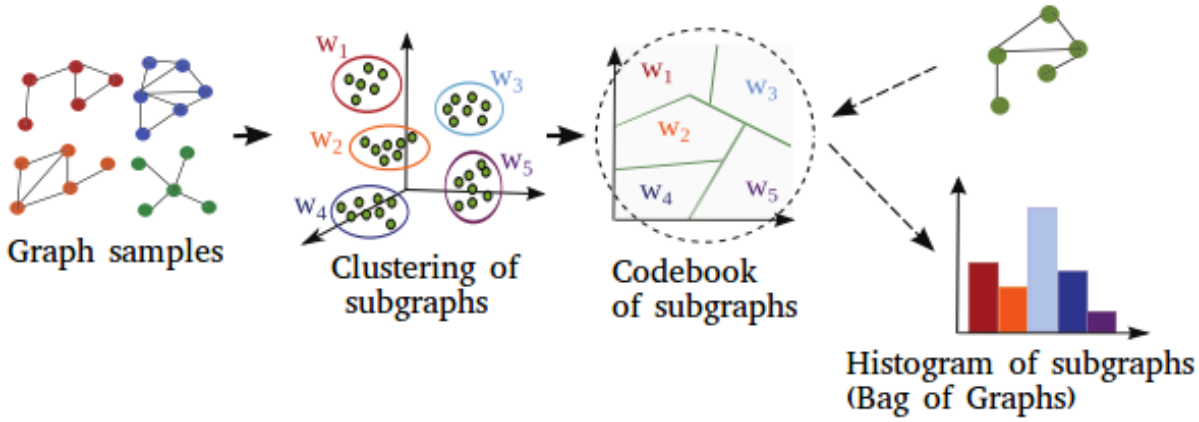


Figure 4.2: Schematic view of the BoG framework for kernel-based graph embedding (adapted from [116]).

$$a_{ij} = \frac{K(D(s_i, w_j))}{sum_{k=1}^{d} K(D(s_i, w_k))} \tag{4.4}$$

$$v_j = \frac{\sum_{i=1}^{|S|} a_{ij}}{|S|} \tag{4.5}$$

$\mathcal{E}_K$ has the potential to produce more discriminative and concise embeddings than $\mathcal{E}_V$ and $\mathcal{E}_H$, but requires additional computation, domain specialization, and adjustment of hyperparameters. Among the three proposed embeddings, $\mathcal{E}_K$ is expected to be the best in terms of effectiveness, but to be the worst with regard to efficiency. This tradeoff must be evaluated in the target scenario to guide the choice of the type of embedding.

4.2.3 Index and Search of Fusion Vectors

Fusion vectors not only act as a representation of multi-ranked objects, but also allows their retrieval, thus promoting intrinsic rank aggregation. This section shows how fusion vectors can be used to promote rank-aggregation by means of an efficient retrieval through indexing and approximate search.

Let $\mathbb{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n\}$ be the fusion vector set related to the response set of a certain collection. A *fusion vector indexer* creates an index of $\mathbb{V}$, in order to allow efficient searches of fusion vectors in response of a query fusion vector, as previously shown in Figure 4.1.

We index fusion vectors by extending the Hierarchical Navigable Small World (HNSW) [89], an archetype that aims approximate K-nearest neighbor searches based on a neighborhood network model. In this work, the authors claimed that a connected graph of a few edges allows efficient searches while retaining high recall rates. We extend the HNSW reference implementation[1] in order to promote indexing and searching of fusion vectors for sparse vectors, and for the cosine dissimilarity.

[1] `https://github.com/nmslib/hnswlib` (As of April, 2019).

4.3 Experimental Evaluation

We present, in this section, the proposed evaluation protocol and the experimental results achieved by our rank aggregation function, in contrast to results from individual rankers and related work.

4.3.1 Datasets and Features

We evaluate the effectiveness and efficiency of our proposal comprising searching scenarios over public datasets of diverse purposes, in order to validate its general applicability. The datasets are listed in Table 4.1, along with the individual rankers adopted to generate ranks for aggregation. The rankers were selected according to their purposes and the objective involved in each dataset. We evaluate the effectiveness of individual rankers, to serve as a first baseline, as well as their rank aggregation with methods proposed in the literature.

UKBench [98] is a dataset of 10,200 images, consisting of 2,550 scenes/objects captured four times each. These captures vary in terms of illumination, viewpoint, and distance. The objects/scenes correspond to the categories, being four samples per class. The effectiveness assessment in UKBench relies on the N-S Score evaluation metric, which varies from zero to four, and measures the mean number of relevant images among the first four images retrieved. We adopt seven rankers, some based on color and texture properties, and also deep-learning-based rankers. In Table 4.1, CNN-Caffe [70] stands for the 4096-dimensional output from the 7th layer of a CNN obtained with the Caffe framework, plus the Euclidean distance as the comparator.

Ohsumed [64] is a bibliographic collection from the National Library of Medicine, consisting of 34,389 abstracts of cardiovascular diseases distributed across 23 categories. We used the subset of 18,302 uni-labeled documents, varying from 56 to 2876 documents per category. We preprocess the documents with stop word removal and Porter's stemming. The rankers adopted comprehend models such as BoW, 2grams, graph-based models along with cosine and Jaccard similarity functions, and language models based on *word embeddings*.

Brodatz [19] is a texture dataset of 1,776 images (texture blocks), being 16 samples for each of the 111 classes (texture types). We adopt three texture rankers.

MPEG-7 [79] is a shape dataset of 1,400 images, distributed in 20 images per 70 categories. We adopt six shape rankers.

Soccer [130] is an image dataset of 280 images, distributed in 40 images per 7 categories. We adopt three color-based rankers.

University of Washington (**UW**) [39] is a hybrid dataset, composed of 1,109 scene pictures annotated by textual keywords, and distributed across 20 classes, varying from 22 to 255 pictures per class. The keywords per picture vary from 1 to 22. We adopt twelve rankers, comprising six textual rankers, three visual color rankers, and three visual texture rankers.

Table 4.1: Datasets and rankers used in the experimental evaluation.

Dataset	Rankers	Ranker Types
UKBench	ACC [67], VOC [135], SCD [91], JCD [146], CNN-Caffe [70], FCTH [28], CEDD [27]	Color, Texture, BoVW, CNN
Ohsumed	BoW-cosine, BoW-Jaccard, 2grams-cosine, 2grams-Jaccard, GNF-MCS [22, 113], GNF-WGU [113, 132], WMD [77]	Textual
Brodatz	LBP [101], CCOM [76], LAS [123]	Texture
MPEG-7	SS [127], BAS [6], IDSC [83], CFD [103], ASC [84], AIR [57]	Shape
Soccer	GCH [122], ACC [67], BIC [121]	Color
UW	GCH [122], BIC [121], JAC [138], HTD [139], QCCH [66], LAS [123], COSINE [9], JACCARD [81], TF-IDF [9], DICE [81], OKAPI [111], BOW [25]	Color, Texture, Textual

4.3.2 Experimental Protocol

The first evaluation intends to analyze the effectiveness of our rank aggregation function in retrieval scenarios, compared to individual rankers and other aggregation functions. Second, we analyze the efficiency and trade-offs of our embedding approaches and indexed formulations.

We perform a protocol for object retrieval, also referred to as ad-hoc retrieval, which is commonly employed to evaluate rank aggregation methods. Given the datasets used, we treat each sample s as query q at a time, whose result candidates belong to S. A retrieved item is relevant to q if they belong to the same class, since we are validating in labeled collections, i.e., relevance scores are either 1 for relevant or 0 for irrelevant. In this case, the query set size corresponds to the dataset size. In general, separate query and response sets can be used, as well as graded relevance.

We measure the retrieval effectiveness with NDCG@10 for all datasets except UKBench, for which we use N-S Score, the standard measure in this dataset.

Three possible combinations of rankers are evaluated per dataset, as in Section 3.3: all rankers; the two most effective rankers; and the pair that maximizes a trade-off measure between high effectiveness and low correlation. We adopt multiple evaluation scenarios per dataset to provide a comprehensive analysis and to allow comparisons of different ranker selection strategies for rank aggregation.

We run and evaluate, using the same experimental procedure, the following baselines: FG (Chapter 3), QueryRankFusion [147], RecKNNGraphCCs [107], RkGraph [106], CorGraph [105], MRA [49], and RRF [32]. For UKBench, we also compare our results to those from the following recent works: Bai and Bai [11], Xie et al. [140], Zheng et al. [151], Zheng et al. [150], Wang et al. [134], and Qin et al. [110]. Other related works could also be included, but we focus on the most recent and competitive unsupervised approaches. In Chapter 3, we demonstrated a number of related works that are no longer competitive to state-of-the-art methods, so that they can be now suppressed in future benchmarks. That comprehends methods such as CombSUM, CombMIN, CombMAX, CombMED, CombANZ, and CombMNZ [52], BordaCount [17], Condorcet, Kemeny, and RLSim [104].

We compare the *winning number* [124] of each rank aggregation function. This allows

us to compare multiple methods concerning several datasets, fusion configurations and baselines. The winning number of a method m, W_m, is a global performance indicator for a performance measure P, expressed by Equation 3.12, where D is the set of datasets, C_d is the set of our 3 pre-defined configurations for dataset d with respect to the rankers to fuse, M is set of rank aggregation methods, $P_m(d, c)$ is the performance of $m \in M$ on $d \in D$ and configuration $c \in C_d$, and $\mathbf{1}_{P_m(d,c)>P_k(d,c)}$ is the indicator function given by Equation 3.13.

The embedding approaches and the indexing scheme, defined before, hold different trade-offs to the retrieval tasks. For this reason, besides effectiveness, we also analyze the efficiency of our method comprising its alternative formulations, and also compare to *FG*, the main baseline. We measure the mean time spent per query, for each combination of rankers in each dataset. The elapsed time for query retrieval refers to the sum of the times spent for fusion graph extraction, fusion vector extraction (graph embedding), and object retrieval, as represented by the online stage in Figure 4.1. The mean time of 5 independent measurements is reported. For the baseline *FG*, the same steps but the fusion vector extraction (absent) is taken into consideration.

In the result reporting and following discussions, we assume the acronyms indicated in Table 4.2.

Table 4.2: Acronyms of the method variants.

Acronym	Meaning
FG	Fusion graph, which is taken as the main baseline for the embedded and indexed approaches of FV
FV-V-COS	Fusion vector embedded by $\mathcal{E}_V$ for cosine dissimilarity
FV-V-JAC	Fusion vector embedded by $\mathcal{E}_V$ for Jaccard dissimilarity
FV-H-COS	Fusion vector embedded by $\mathcal{E}_H$ for cosine dissimilarity
FV-H-JAC	Fusion vector embedded by $\mathcal{E}_H$ for Jaccard dissimilarity
FV-K-COS	Fusion vector embedded by $\mathcal{E}_K$ for cosine dissimilarity
FV-K-JAC	Fusion vector embedded by $\mathcal{E}_K$ for Jaccard dissimilarity
FV-V-COS-FAST	The indexed counterpart of FV-V-COS
FV-H-COS-FAST	The indexed counterpart of FV-H-COS
FV-K-COS-FAST	The indexed counterpart of FV-K-COS

4.3.3 Ranker Effectiveness

The effectiveness of the rankers are shown in Tables 4.3a, 4.3b, 4.3c, 4.3d, 4.3e, and 4.3f for the datasets Brodatz, UW, MPEG-7, Ohsumed, UKBench, and Soccer, respectively.

These results serve as an initial baseline for the rank aggregation functions, so that the aggregation functions are expected to overcome them.

We can observe large variability in rankers' results. Rankers' relative performance also vary depending on the dataset, thus providing complementary views. JACCARD, for instance, was superior to COSINE in UW dataset, but was worse in Ohsumed.

Table 4.3: Effectiveness of individual rankers on the datasets.

(a) Brodatz

Ranker	NDCG@10
LAS	**0.850533**
CCOM	0.726186
LBP	0.652759

(b) UW dataset

Ranker	NDCG@10
JAC	**0.810729**
BIC	0.746454
DICE	0.722831
BOW	0.720781
OKAPI	0.716035
JACCARD	0.701651
TF-IDF	0.658880
GCH	0.630315
COSINE	0.554767
LAS	0.514314
HTD	0.495002
QCCH	0.414249

(c) MPEG-7

Ranker	NDCG@10
ASC	**0.941585**
AIR	0.939424
CFD	0.930685
IDSC	0.922828
BAS	0.866098
SS	0.611481

(d) Ohsumed

Ranker	NDCG@10
BoW-cosine	**0.669701**
2grams-cosine	0.664120
GNF-WGU	0.662668
GNF-MCS	0.655420
2grams-Jaccard	0.651320
BoW-Jaccard	0.645711
WMD	0.427361

(e) UKBench

Ranker	N-S Score
VOC	**3.54**
ACC	3.37
CNN-Caffe	3.31
SCD	3.15
JCD	2.79
FCTH	2.73
CEDD	2.61

(f) Soccer

Ranker	NDCG@10
BIC	**0.614818**
ACC	0.592699
GCH	0.536412

4.3.4 Rank Aggregation Results

The effectiveness of our method, for different embedding approaches and indexed formulations, along with the related works, are shown in Tables 4.4, 4.5, 4.6, 4.7, 4.8, and 4.9, respectively for UKBench, Ohsumed, Brodatz, MPEG-7, Soccer, and UW. Three ranker selections are evaluated per dataset.

The choice between cosine or Jaccard dissimilarity as the comparator for the fusion vectors did not show a clear winner, regarding non-indexed formulations. Besides, as our indexed formulations are currently implemented only with the cosine dissimilarity, that is for now the preferred choice for comparing fusion vectors.

FV-K was the most effective rank aggregation function, compared to *FV-V* and *FV-H*, in 4 of 6 datasets. As expected, the kernel-based embedding performed better than the other two approaches, but at a higher computational cost. Interestingly, *FV-V* was better than *FV-H* in 4 of 6 datasets, which may be due to its large increase in vector dimensionality.

FV overcame *FG*, the strongest and main baseline, in all 6 datasets but Ohsumed. Nevertheless, in Ohsumed, *FV* surpassed all other baselines. As *FV-K* did not perform well in Ohsumed and UKBench, we conjecture that our graph kernel structure and clustering selection criteria for *FV-K* should be specialized per domain, even though they went well in most cases. For example, the kernels for BoG could be defined by larger

Table 4.4: FV Results for rank aggregation on UKBench.

Method	N-S Score		
	VOC + ACC + CNN-Caffe + SCD + JCD + FCTH + CEDD	VOC + ACC	VOC + ACC + CNN-Caffe
FV-V-COS-FAST	3.74	3.86	3.92
FV-H-COS-FAST	3.74	3.86	3.92
FV-V-COS	3.74	3.86	3.92
FV-H-COS	3.74	3.86	3.92
FV-V-JAC	3.69	3.84	3.90
FV-H-JAC	3.69	3.84	3.90
FG	3.69	3.83	3.90
RecKNNGraphCCs	3.67	3.81	3.87
QueryRankFusion	3.60	3.78	3.86
FV-K-COS-FAST	3.60	3.72	3.81
FV-K-COS	3.60	3.72	3.81
MRA	3.52	3.50	3.77
RRF	3.52	3.60	3.76
FV-K-JAC	3.50	3.56	3.72
RkGraph	3.03	3.50	3.54
CorGraph	2.44	2.91	2.77

paths within the graphs, in it was explored in [44]. We leave this investigation for future work.

In UKBench, Brodatz, MPEG-7, Soccer and UW, the indexed versions of all the three embeddings in all three aggregation scenarios promoted nearly the same effectiveness results than their non-indexed versions. In Ohsumed, the indexed versions of FV-V and FV-H had nearly equivalent effectiveness to their non-indexed versions, and only for FV-K the indexing actually decreased the results (by 4%) but only in one of three aggregation scenarios. Regarding the effectiveness of the indexed formulations for FV, we can conclude that the indexing effectively contributes to our solution in terms of efficiency, while retaining the quality results in almost every case.

By comparing the aggregation functions globally in terms of winning numbers, *FV* overcame all baselines, if we take the best result of its variants per aggregation per dataset, as shown in Figure 4.3.

When we compare each *FV* variant separately, along with the baselines, the best *FV* approach is competitive and overcome all baselines but *FG*, in terms of effectiveness (Figure 4.4). This is due to the unsupervised nature of the problem, which does not involve hyperparameter adjustment. The *FV* approaches are competitive between each other, varying in which performs best per dataset. This issue, however, can be solved by additional pre-validation steps, if desired, in order to pick the most promising embedding. Besides, the efficiency benefits of our method can be of critical importance depending on the task. We investigate this trade-off in the following section.

Table 4.5: FV Results for rank aggregation on Ohsumed.

Method	NDCG@10		
	BoW-cosine + BoW-Jaccard + 2grams-cosine + 2grams-Jaccard + GNF-MCS + GNF-WGU + WMD	BoW-cosine + 2grams-cosine	BoW-cosine + WMD
FG	0.683835	0.683472	0.676760
FV-V-JAC	0.681238	0.676525	0.644981
FV-H-JAC	0.681220	0.680822	0.673146
RecKNNGraphCCs	0.676234	0.679728	0.667750
FV-H-COS-FAST	0.676325	0.678596	0.664745
FV-H-COS	0.676310	0.678419	0.665320
FV-V-COS-FAST	0.666759	0.672709	0.624808
FV-V-COS	0.666511	0.672674	0.627062
QueryRankFusion	0.651279	0.671258	0.669704
MRA	0.666045	0.670357	0.582049
RRF	0.665793	0.671294	0.571016
FV-K-COS	0.596524	0.665829	0.502955
FV-K-JAC	0.469095	0.674610	0.432484
FV-K-COS-FAST	0.545134	0.666344	0.502555
CorGraph	0.487177	0.497431	0.456434
RkGraph	0.289045	0.688443	0.288436

4.3.5 Efficiency Analysis

Figures 4.5, 4.6, 4.7, 4.8, 4.9, and 4.10 present the effectiveness scores related to the mean query times, for the three rank aggregations, in UKBench, Ohsumed, Brodatz, MPEG-7, Soccer, and UW, respectively. The times were measured on an Intel Core i7-7500U CPU @ 2.70GHz with 16GB of RAM.

FV held much lower query times than *FG* in all datasets, while preserving or surpassing its effectiveness in 4 of 6 datasets. Speedups from 10x to 100x were achieved by the indexed formulations. The gains are more significant for larger datasets, while also more demanded in those cases. Even for non-indexed FV approaches, the query times are considerably lower than *FG*.

The efficiency for *FV*, regardless of their indexed or non-indexed formulations, are affected by the number of dimensions, specifically the number of non-zero entries in the resulting fusion vectors as the dissimilarity functions can be designed for sparse vectors. In the non-indexed formulations, the collection size is the critical aspect for the final retrieval times, although also affected by the dimensionality. In general, FV-V is the approach that conducts to the lowest dimensionality. FV-H presents high dimensionality, but actually not that high in terms of non-zero entries. FV-K, in practice, is the embedding that produced the fusion vectors with higher non-zero dimensions, as illustrated in Table 4.10 for some evaluation scenarios. That explains the lower efficiency in its retrieval when compared to the other two approaches, although it is still much better than non-indexed *FV* formulations and *FG*, and also presents some effectiveness gains over its alternatives.

Table 4.6: FV Results for rank aggregation on Brodatz.

Method	NDCG@10		
	LAS+CCOM+LBP	LAS+CCOM	LAS+LBP
RecKNNGraphCCs	0.877882	0.882903	0.839717
FV-K-COS-FAST	0.883388	0.880668	0.839430
FV-K-COS	0.883388	0.880666	0.839425
FV-K-JAC	0.881201	0.879870	0.838423
FG	0.878995	0.872084	0.835624
FV-H-JAC	0.879220	0.872057	0.835416
FV-H-COS-FAST	0.867736	0.858995	0.825835
FV-H-COS	0.867688	0.858960	0.825785
FV-V-JAC	0.877373	0.868278	0.830557
FV-V-COS-FAST	0.863513	0.854987	0.821171
FV-V-COS	0.863513	0.854909	0.821135
RkGraph	0.812659	0.861250	0.788682
QueryRankFusion	0.850263	0.850438	0.808562
RRF	0.818656	0.817139	0.788840
MRA	0.822778	0.813396	0.788883
CorGraph	0.749420	0.895623	0.719204

The experimental results achieved by FV concerning effectiveness and efficiency shows a solid evidence that *fusion vectors* yield comparable or superior performance when compared with the state-of-the-art rank aggregation functions in effectiveness, while providing a fast alternative for aggregating lists in search systems, a common shortcoming from previous works.

4.4 Conclusions

We introduced the concepts of embedding and indexing of graph-based rank aggregation representations, and their application for search systems.

Unsupervised embedding formulations were proposed and discussed, based on vertices, a hybrid of vertices and edges, and kernels. The concept of fusion vectors was introduced, based on which a retrieval model could be established. The possibility of representing contextual information defined in terms of multiple ranks into a vector, opened the possibility of exploring indexing schemes. We also investigated the use of approximate searches to deliver fast retrieval based on rank aggregation.

We demonstrated the flexibility of the proposed method in multimodal retrieval tasks, and evaluated the method experimentally across many diverse search scenarios, considering comparison with multiple state-of-the-art baselines. Conducted experiments showed that our approach leads to comparable or superior results when compared with start-of-the-art rank aggregation functions considering effectiveness, while bringing a novel approach for fast retrieval on that context. The efficiency analysis showed a speedup improvement from 10 to 100 against our strongest baseline. An extensive experimental section was conducted, considering 7 recent related works, in 3 distinct scenarios for each of the 6 datasets investigated.

In a future work, we plan to address the following research directions: (i) the investigation of alternative graph-based rank aggregations within our methodology, (ii) supervised

Table 4.7: FV Results for rank aggregation on MPEG-7.

Method	NDCG@10		
	AIR + CFD + ASC + IDSC + BAS + SS	ASC + AIR	AIR + CFD
FV-K-JAC	0.997652	0.997563	0.998515
FV-K-COS-FAST	0.998322	0.997401	0.998182
FV-K-COS	0.998322	0.997401	0.998182
RecKNNGraphCCs	0.998052	0.995160	0.997267
FV-V-JAC	0.997997	0.995432	0.995265
FV-H-JAC	0.997892	0.994871	0.995817
FG	0.997658	0.994729	0.995886
FV-H-COS-FAST	0.996635	0.991278	0.991684
FV-H-COS	0.996635	0.991278	0.991684
FV-V-COS	0.996272	0.988776	0.987672
FV-V-COS-FAST	0.996272	0.988776	0.987671
RkGraph	0.826119	0.999350	0.992078
CorGraph	0.992456	0.962951	0.961460
RRF	0.980638	0.957684	0.954499
MRA	0.980086	0.950442	0.946144
QueryRankFusion	0.940976	0.941762	0.941271

or semi-supervised approaches for fusion graph generation and embedding, and (iii) the validation of fusion graphs and fusion vectors in other tasks, such as recommendation or classification.

Table 4.8: FV Results for rank aggregation on Soccer.

Method	NDCG@10		
	BIC+ACC+GCH	BIC+ACC	BIC+GCH
RkGraph	0.653623	0.656422	0.628563
FV-K-COS-FAST	0.642172	0.656412	0.628528
FV-K-COS	0.641453	0.656173	0.628474
FG	0.651828	0.655332	0.622217
FV-H-JAC	0.651278	0.652244	0.621678
FV-H-COS-FAST	0.651308	0.650073	0.621168
FV-K-JAC	0.642329	0.649922	0.626852
FV-H-COS	0.650997	0.649573	0.620213
FV-V-JAC	0.649043	0.652008	0.620128
FV-V-COS-FAST	0.647671	0.648442	0.619718
FV-V-COS	0.647149	0.648151	0.619468
CorGraph	0.645004	0.643505	0.623627
RecKNNGraphCCs	0.637537	0.640729	0.618704
QueryRankFusion	0.613732	0.613659	0.598862
RRF	0.604119	0.613005	0.590819
MRA	0.605971	0.611017	0.588399

Table 4.9: FV Results for rank aggregation on UW.

Method	NDCG@10		
	JAC + BIC + DICE + BOW + OKAPI + JACCARD + TF-IDF + GCH + COSINE + LAS + HTD + QCCH	JAC + BIC	JAC + OKAPI
CorGraph	0.896341	0.842665	0.933452
FV-K-JAC	0.890008	0.862473	0.909473
FV-K-COS-FAST	0.888936	0.857554	0.899520
FV-K-COS	0.888770	0.857554	0.899256
FV-V-JAC	0.875202	0.850045	0.886401
FV-H-JAC	0.874106	0.853056	0.884608
FG	0.873607	0.854473	0.882776
RecKNNGraphCCs	0.869448	0.843423	0.882035
FV-V-COS-FAST	0.862406	0.841015	0.877076
FV-H-COS-FAST	0.862289	0.844613	0.874147
FV-H-COS	0.862108	0.844539	0.874078
FV-V-COS	0.862229	0.840928	0.877016
RkGraph	0.746804	0.841127	0.866544
MRA	0.815983	0.797292	0.786995
RRF	0.815779	0.798502	0.795143
QueryRankFusion	0.747281	0.792681	0.807250

Table 4.10: Dimensionality for each embedding approach. For FV-V and FV-H, we report the theoretical and occupied (non-zero) dimensions.

Dataset	Rankers	FV-V		FV-H		FV-K
		occupied	limit	occupied	limit	
MPEG-7	ASC + AIR	13.76 ± 1.34	1,400	92.19 ± 12.99	1,961,400	19.71 ± 7.88
Ohsumed	BoW-cosine + WMD	16.37 ± 1.01	18,302	67.09 ± 10.92	334,981,506	899.65 ± 388.73
Soccer	BIC + GCH	14.14 ± 1.38	280	59.45 ± 8.40	78,680	131.56 ± 31.21

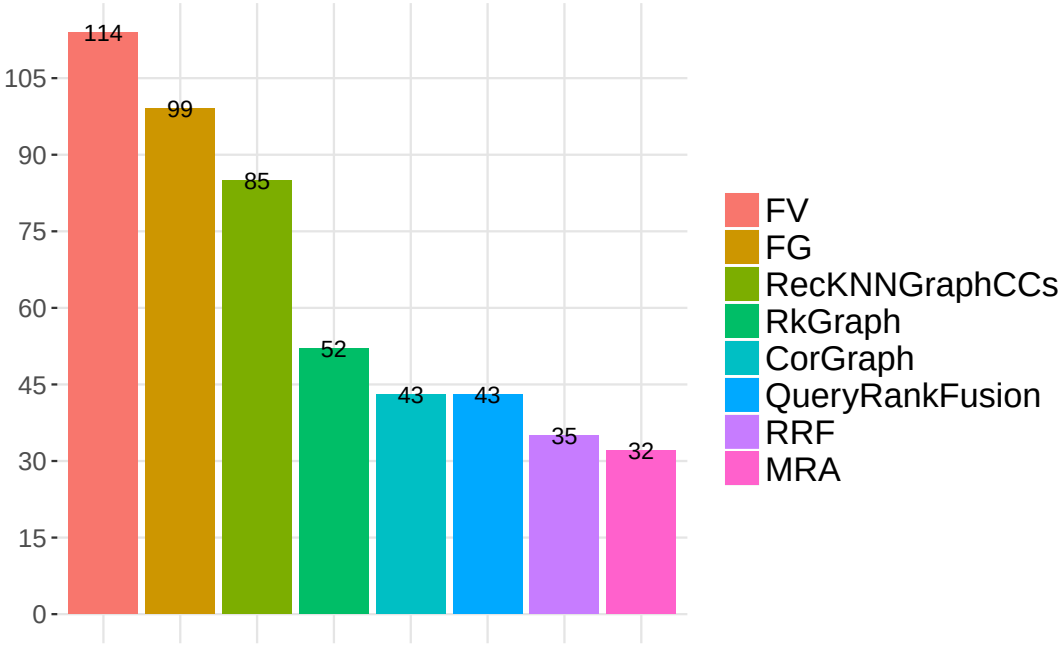


Figure 4.3: Winning numbers achieved per rank aggregation function.

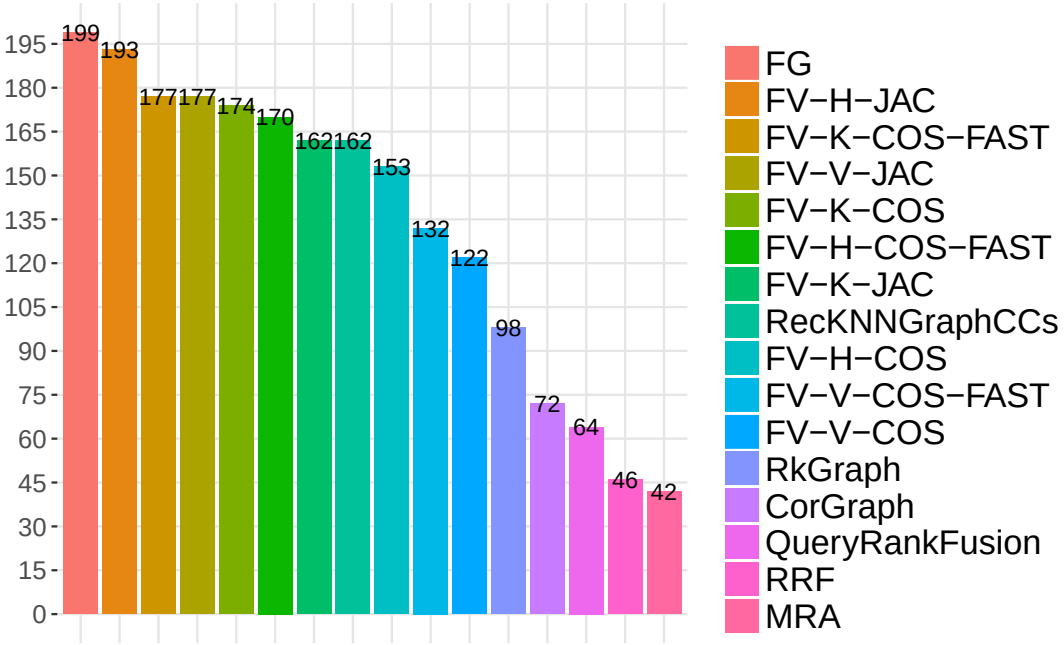


Figure 4.4: Winning numbers achieved per rank aggregation function, including all FV possible approaches.

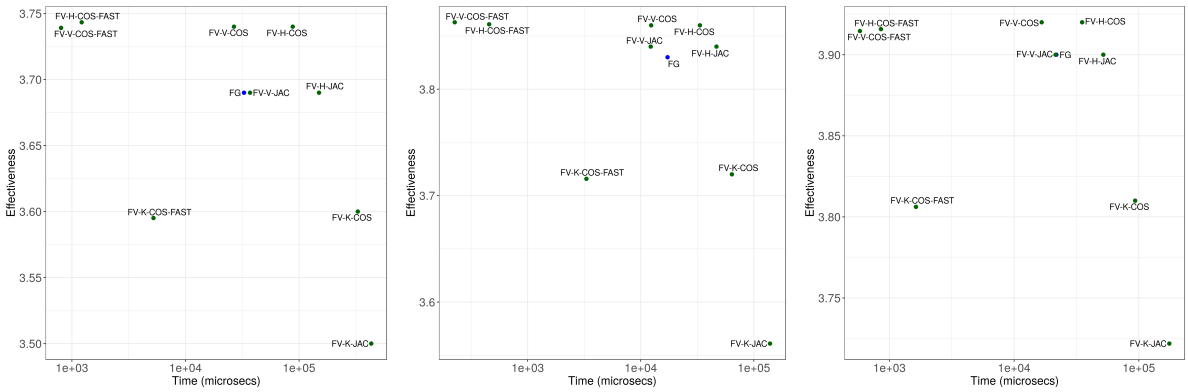


Figure 4.5: Effectiveness and efficiency trade-offs for FV and its embedding and indexed versions, in UKBench, for VOC + ACC + CNN-Caffe + SCD + JCD + FCTH + CEDD, VOC + ACC, and VOC + ACC + CNN-Caffe, respectively.

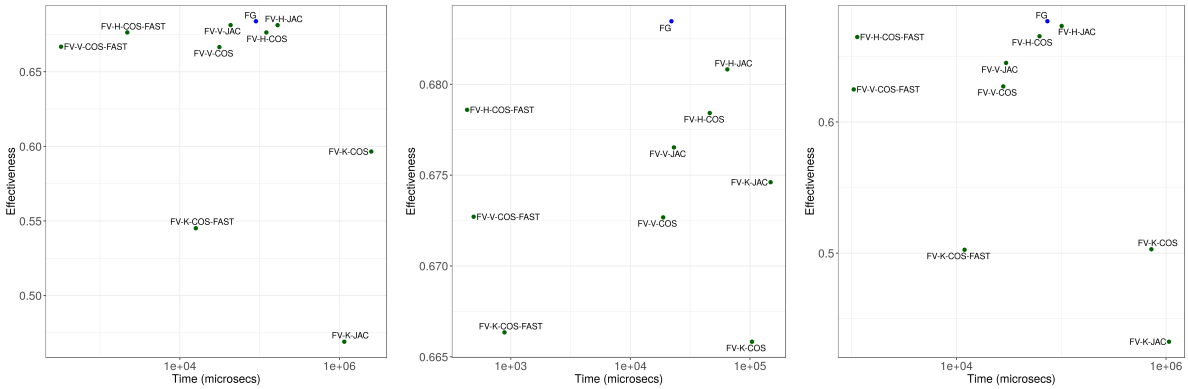


Figure 4.6: Effectiveness and efficiency trade-offs for FV and its embedding and indexed versions, in Ohsumed, for BoW-cosine + BoW-Jaccard + 2grams-cosine + 2grams-Jaccard + GNF-MCS + GNF-WGU + WMD, BoW-cosine + 2grams-cosine, and BoW-cosine + WMD, respectively.

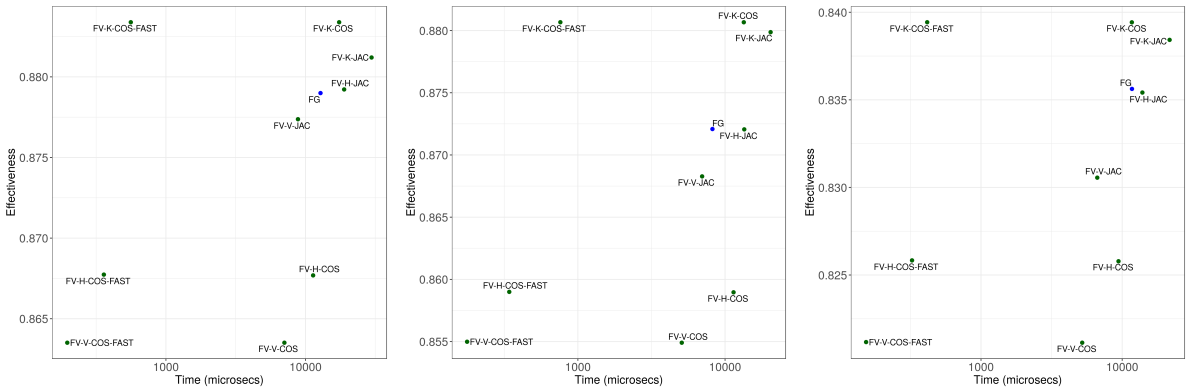


Figure 4.7: Effectiveness and efficiency trade-offs for FV and its embedding and indexed versions, in Brodatz, for LAS + CCOM + LBP, LAS + CCOM, and LAS + LBP, respectively.

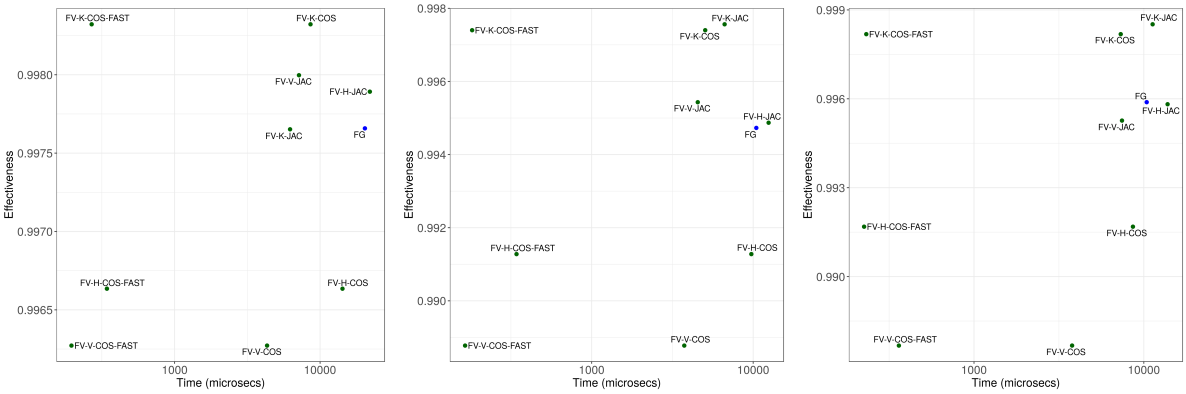


Figure 4.8: Effectiveness and efficiency trade-offs for FV and its embedding and indexed versions, in MPEG-7, for AIR + CFD + ASC + IDSC + BAS + SS, ASC + AIR, and AIR + CFD, respectively.

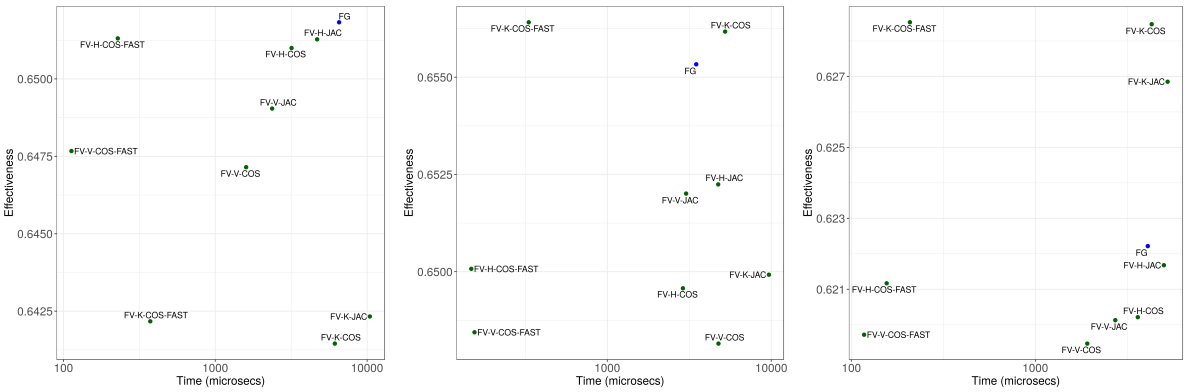
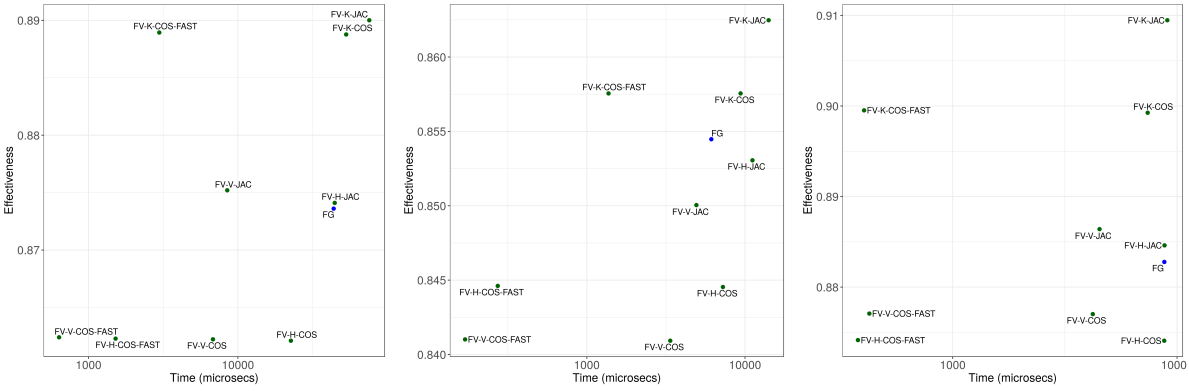


Figure 4.9: Effectiveness and efficiency trade-offs for FV and its embedding and indexed versions, in Soccer, for BIC + ACC + GCH, BIC + ACC, and BIC + GCH, respectively.



(a) (b) (c)

Figure 4.10: Effectiveness and efficiency trade-offs for FV and its embedding and indexed versions, in UW, for (a) JAC + BIC + DICE + BOW + OKAPI + JACCARD + TF-IDF + GCH + COSINE + LAS + HTD + QCCH, (b) JAC + BIC, and (c) JAC + OKAPI.

Chapter 5

Multimodal Graph-Based Rank Fusion Representation for Prediction

5.1 Introduction

Most previous initiatives for multimodal prediction are solely based on either CNN-based descriptors in isolation, feature concatenation [15, 37, 54, 60, 86, 125], or graph-based feature-fusion [7, 137]. These approaches still ignore the correlation between modalities, as well as object correlation, and they are not consistently better than ranking models that do not rely on fusion.

Despite most real scenarios that do not contain or can not afford labeled data, unsupervised learning still needs more investigation in the literature, specially for multimodal representation models. We explore how unsupervised rank aggregation capabilities can be applied to prediction tasks. We claim that unsupervised rank aggregation functions can provide an effective dataset exploitation. This work presents an unsupervised representation model, based on rank-fusion graphs, for general applicability in multimodal prediction tasks, such as classification and regression. We explore and extend the concept of a rank-fusion graph, that was originally proposed as part of a rank aggregation approach for retrieval tasks, in Chapter 3.

We present a fusion method based on the representation of multiple ranks, defined according to different criteria, into a graph. Graphs provide an efficient representation of arbitrary structures and inter-relationships among different elements of a model. We embed the generated graph into a feature space, creating fusion vectors. This approach is able to learn and encode the intrinsic manifold from the collection automatically, without any supervision. Next, an estimator is trained to predict if an input multimodal object refers to a target label (or event) or not, following their fusion vectors.

In Chapter 3, a methodology to apply rank-fusion graphs for efficient retrieval was presented, in the context of retrieval tasks. Here we explore those graph embedding approaches of rank-fusion graphs, now targeting a representation model for prediction tasks, either supervised or unsupervised. For this purpose, we propose specific components for the training and prediction phases, as well as a new application for the fusion vectors.

As we propose a representation model, our solution can be seen as an early-fusion

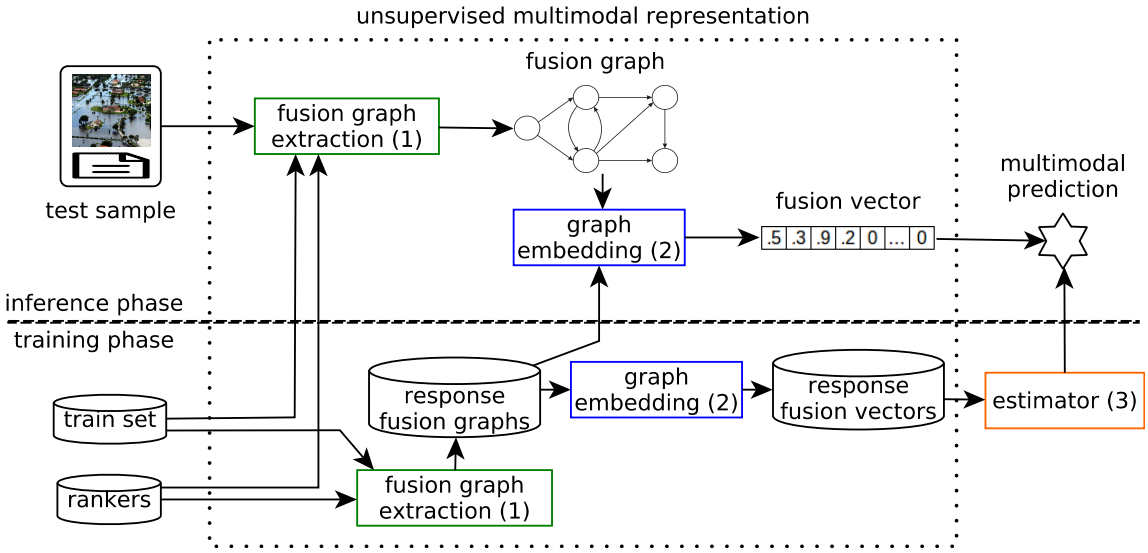


Figure 5.1: Proposed graph-based rank fusion for multimodal prediction.

approach. Nevertheless, it is based on retrieval models, without the need to work directly on feature level. In this sense, we categorize the method as hybrid.

Our method has the advantage of being unsupervised. It also explores and captures relationships from the collection into the representation model, and it works on top of any descriptors for multimodal data, such as visual or textual. By promoting a representation model solely based on base descriptors and unsupervised data analysis over the collection, we conjecture that our approach leads to a competitive multimodal representation model that explores and encodes information from multiple descriptors and underlying sample relationships automatically, while not requiring labeled data.

Experimental results over multiple multimodal and visual datasets demonstrate that the proposal is robust for different detection scenarios involving textual, visual, and multimodal features, yielding better detection results than state-of-the-art methods from both early-fusion and late-fusion approaches.

We introduce the notion of a representation model from rank-fusion graphs, and demonstrate its application for multimedia flood detection. We propose and discuss alternative approaches for the representation model. We also evaluate the method extensively over multiple multimodal and image scenarios, to analyze its applicability for prediction tasks in general. We evaluate the method against early-fusion and late-fusion approaches.

5.2 Representation and Prediction Based on Graph-Based Rank Fusion

Figure 5.1 presents an overview of our method – a multimodal representation and estimator based on rank-fusion graphs. The solution is composed of three main generic components, briefly described here and detailed in the following sections. The multimodal representation is completely unsupervised, thus able to be adopted in any tasks in the absence of labeled data.

Two phases are defined. The *training* phase comprehends the modeling of the train set in terms of multiple rankers, as fusion graphs and then as fusion vectors. This step performs a graph embedding of fusion graphs and the training of a multimodal prediction model. The *inference* phase refers to the multimodal prediction preceded by a rank-based fusion approach for multimodal representation. The training phase is performed only once, while the inference phase is performed per prediction. The first two components – *fusion graph extraction* and *graph embedding* – are used in both phases.

The *fusion graph extraction* (component 1 in the figure) generates a fusion graph $\mathcal{G}$ for a given test sample q. $\mathcal{G}$ consists of an aggregated representation of multiple ranks for q, thus capturing and correlating information of multiple ranks. This formulation is presented in Section 5.2.1. *Graph Embedding* (2) projects fusion graphs into a vector space model, producing a corresponding fusion vector $\mathcal{V}$ for $\mathcal{G}$. We present the embedding formulation in Section 5.2.2. At the end, an *estimator* (3) is built based on the response fusion vectors, in order to predict for test samples (also modeled as fusion vectors). This component is detailed in Section 5.2.3.

5.2.1 Rank-Fusion Graphs

This component produces a fusion graph $\mathcal{G}$ for a given query sample q, also in terms of m rankers and n response items. A fusion graph is a graph-based encoding of multiple ranks for q, that encapsulates and correlates ranks.

We follow the fusion graph formulation from Chapter 3, referred to as FG, that defines a mapping function $q \mapsto \mathcal{G}$, based on its ranks $\tau \in \mathcal{T}_q$ and ranks' inter-relationships. The proposed formulation also includes a dissimilarity function for $\mathcal{G}$, and a retrieval model based on fusion graphs. Here, however, we focus on the definition of $\mathcal{G}$, extending its use as part of a rank-based late-fusion approach for representation model in prediction tasks, without these components.

The process is illustrated in Figure 3.2. Given a query (or test sample) q, m rankers, and a training set of size n, m ranks are generated. These ranks are then normalized to allow for producing the fusion graph $\mathcal{G}$ for q. In rank normalization, the scores in the ranks generated by dissimilarity-based comparators are converted to similarity-based scores. Besides, all ranks have their scores rescaled to the same interval. $\mathcal{G}$, for q, includes all response items from each $\tau_q \in \mathcal{T}_q$, as vertices. Vertices are connected by taking into account the degree of relationship between their corresponding response items, and the degree of their relationships to q.

5.2.2 Embedding of Rank-Fusion Graphs

Let $\mathbb{G} = \{\mathcal{G}_i\}_{i=1}^{n}$ be the fusion graph set related to the response set of a given collection. Based on $\mathbb{G}$, an embedding function $\mathcal{E}$ defines a vector space model in order to project a fusion graph $\mathcal{G}$ into that space as a fusion vector $\mathcal{V}$, i.e. $\mathcal{V} = \mathcal{E}(\mathcal{G})$ for any $\mathcal{G}$.

We investigate how $\mathcal{V}$ can be adopted as a representation model of multi-ranked objects. It encodes the use of multiple rankers and allows the fusion of multiple modalities, being therefore suitable for prediction tasks.

$\mathcal{E}$ can be defined by unsupervised or supervised approaches. We explore three approaches in this work, preliminarly presented in Section 4.2.2 in the context of retrieval tasks. Different from that work, here we focus its use on prediction tasks involving supervised learning.

This first one, $\mathcal{E}_V$, derives the vector space based on vertex analysis. Let $w(v)$ be the weight of the vertex v, if $v \in \mathcal{G}$, otherwise 0. Similarly, let $w(e)$ be the weight of the edge e, if $e \in \mathcal{G}$, otherwise 0. Also, let N be the dimensionality of the vector space model defined by $\mathcal{E}$, such that $\mathcal{V} \in \mathbb{R}^N$. $\mathcal{E}_V$ derives $\mathcal{V}$ from the vertices of $\mathcal{G}$. There is one vector attribute relative to each response object, therefore $N = |\mathbb{G}|$. $\mathcal{V}$ is derived from $\mathcal{G}$ such that $|\mathcal{V}| = N$, $\mathcal{V}[i]$ is the importance value of the i-th attribute, and $\mathcal{V}[i] = w(v_i)$. Despite the vector space increases linearly to the collection size, the resulting fusion vectors are mainly sparse, i.e., composed of few non-zero entries, which makes this embedding formulation simple and efficient in practice.

$\mathcal{E}_H$ is a hybrid embedding approach based on both vertex and edge analysis. $\mathcal{E}_H$ encodes more information into the vector space, at a cost of a higher dimensionality.

The third approach, $\mathcal{E}_K$, extends the BoG [116] archetype to embed graphs as a histogram of kernels, where the vectorial attributes are selected by unsupervised selection of common subgraph patterns. The kernels are obtained from the centroids of a graph clustering process. Then, a vector quantization process, consisting of assignment and pooling procedures, is adopted to embed an input graph to the vector space. We adopt SOFT assignment and AVG pooling, as in Section 4.2.2. BoG has been successfully applied in scenarios involving graph classification, textual representation and information retrieval [43, 44, 116]. To the best of our knowledge, this is the first extension of BoG in the context of multimodal representation.

We refer to FV-V as the fusion vector generated by $\mathcal{E}_V$, while FV-H is generated by $\mathcal{E}_H$, and FV-K by $\mathcal{E}_K$.

5.2.3 Prediction based on Fusion Vectors

Fusion vectors allow the creation of predictors, such as classifiers or regressors, and also ad-hoc retrieval systems, depending on the underlying demanded task. In this work, we adopt them to build predictors, where training objects – associated with ground-truth information – are used to train an estimator for a certain input object be considered a label (or event) or not.

Let S be a training corpus of size n. A predictor can be modeled as $f(X, \beta) \approx Y$, where f is an approximation function, X are the independent variables, Y is the dependent variable (target), and β are unknown parameters. A learning model explores S to find a f that minimizes a certain error metric. The training samples are generally labeled, so Y may be categorical. Still, a regressor can be built, as $E(Y|X) = f(X, \beta)$, so that posterior probabilities are inferred in order to estimate a confidence of a sample to refer to a class of not. In our case, X refers to the fusion vectors, acting as variables that describes the samples in terms of their multiple multimodal ranks. For Y, we adopt the categorical labels from the training set.

5.3 Experimental Evaluation

We present, in this section, the experimental protocol used to evaluate the method, and the results achieved comparatively to state-of-the-art baselines. We evaluate the effectiveness of our method as a representation model in prediction tasks. The focus is on validating our fusion method comparatively to the individual use of descriptors with no fusion, as well as to compare it to early-fusion and late-fusion approaches.

5.3.1 Evaluation Scenarios

We evaluate the proposed method on multiple datasets, comprised of heterogeneous multimodal data, in order to assess its general applicability. Our experimental evaluation comprises the following scenarios:

- **ME17-DIRSM** dataset, the acronym for MediaEval 2017 Disaster Image Retrieval from Social Media [16], is a multimodal dataset of a competition whose goal is to infer whether images and/or texts refer to flood events or not. The samples contain images along with textual metadata, such as title, description, and tags, and they are labeled as either flood (1) or non-flood (0). The task predefines a development set (devset) of 5,280 samples, and a test set of 1,320 samples, as well as its own evaluation protocol.

- **Brodatz** [19] is a dataset of texture images, labeled across 111 classes. There are 16 samples per class, composing a total of 1,776 samples.

- **Soccer** [130] is an image dataset, labeled across 7 categories (soccer teams), containing 40 images each.

- **UW** [39], also called University of Washington dataset, is a multimodal collection of 1,109 images annotated by textual keywords. The images are pictures labeled across 22 classes (locations). Pictures per class vary from 22 to 255. The number of keywords per picture vary from 1 to 22.

5.3.2 Evaluation Protocol

For any dataset that does not explicitly define train and test sets, we initially split it in train and test sets, at a proportion of 80% and 20% respectively, in a stratified way so that the proportions per class remain equal. The same train and test sets per dataset are adopted to evaluate all methods under the same circumstances, as well as the evaluation metrics.

For each representation model, we fit a multiclass SVM classifier, with one-vs-all approach and linear kernel, as it is a good fit for general applicability. Hyper-parameters are selected by grid search on the train set, using an internal 5-fold cross validation.

We evaluate the effectiveness of each method by the balanced accuracy score, which is suitable to evaluate on either balanced and imbalanced datasets. The methods are compared by their balanced accuracy.

The descriptors compose rankers, which are employed to generate ranks in our late-fusion representation. Our method varies with respect to which rankers are used, whether visual or textual rankers, or even their combinations for the multimodal scenario are applied. Besides that, it also varies with respect to which embedding approach is adopted. We evaluate these aspects experimentally.

We model our solution as a rank-fusion approach, followed by an estimator based on rank-fusion vectors. This approach intends to validate our hypothesis that unsupervised graph-based rank-fusion approaches can lead to effective representation models for prediction tasks in general.

We adopt the same experimental evaluation for all datasets but ME17-DIRSM, that defines its own procedure. In this case, the task imposes three evaluation scenarios, as follows. In the first one, called "visual", only visual data can be used. In the second scenario, called "textual", only textual data are used. In the third scenario, called "multimodal", both visual and textual data are expected to be used. The correctness is evaluated, over the test set, by the metric Average Precision at K (AP@K) at various cutoffs (50, 100, 250, 480), and by their mean value (mAP).

Although the ME17-DIRSM task may be seen as a multimodal binary classification problem, the evaluation metrics require ranking-based solutions, or equivalently confidence-level regressors, so that the first positions are the most likely to refer to a flood event. For the estimator component in ME17-DIRSM, we adopt SVR, an L2-regularized logistic regression based on linear SVM in its dual form, with probabilistic output scores, and trained over the fusion vectors from devset. Probabilistic scores are used so that we can sort the test samples by confidence expectancy of being flood.

In ME17-DIRSM, our results are compared to those from state-of-the-art baselines. In Soccer, Brodatz, and UW, our results are compared to those from two major fusion approaches: concatenation, and majority vote. They cover baselines from both early-fusion and late-fusion families. For the concatenation procedure, besides the concatenation itself, we normalize the vectors to the $[0, 1]$ interval for each attribute, in order to avoid disparities due to different descriptor attribute ranges. We apply majority voting in the scenarios involving an odd number of descriptors, so that each predicted class is taken as the one most frequently predicted by the estimators constructed for each descriptor.

5.3.3 Descriptors and Rankers

The ME17-DIRSM dataset provides one image per sample, along with pre-extracted feature vectors by 9 classical image content descriptors, such as ACC [67] and CEDD [27]. Despite the visual features provided, we ended up choosing descriptors based on Convolution Neural Networks (CNN). In preliminary analysis, we noticed that CNN-based descriptors surpassed most classical descriptors by large margins. We selected three visual descriptors and three textual descriptors, for individual analysis in the designed evaluation scenarios, and to evaluate different possibilities of rank-fusion aggregations. We adopt the following state-of-the-art visual descriptors:

- *ResNet50IN*: 2048-dimensional average pooling of the last convolutional layer of ResNet50 [62], pre-trained on ImageNet [112], a dataset of about 14M images labeled

for object recognition;

- *VGG16P365*: 512-dimensional average pooling of the last convolutional layer of VGG16 [117], pre-trained on Places365-Standard [152], a dataset of about 10M images of labeled scenes;
- *NASNetIN*: 2048-dimensional average pooling of the last convolutional layer of NASNet [155], pre-trained on ImageNet dataset.

Based on the textual metadata in ME17-DIRSM, we adopt the following descriptors:

- *BoW*: Bag of Words (BoW) with TF weighting;
- *2grams*: 2grams with TF weighting;
- *doc2vecWiki*: 300-dimensional doc2vec [80] pre-trained on English Wikipedia dataset, of about 35M documents and dumped at 2015-12-01.

For the other datasets, we elected a number of heterogeneous descriptors:

- Soccer: BIC, GCH, and ACC.
- Brodatz: JCD, FCTH, and CCOM.
- UW: JAC, ACC, JCD, and CEDD, as visual descriptors, and *word2vecSum*, *word2vecAvg*, *doc2vecWiki*, and *doc2vecApnews*, as textual descriptors.

For the deep networks used for CNN-based visual feature extraction, as well as in the textual feature extraction with doc2vec, we take advantage of pre-trained models. This practice, known as transfer learning, has been effective in many scenarios [74], and it is also particularly beneficial for datasets that are not large enough to generalize the training of such large architectures, as in our case. Because the problem requires prediction of flood images, we prioritize, in the selection of visual descriptors, datasets for pre-training that focus on images of scenes, aiming at better generality to the target problem.

We perform the same preprocessing steps for every textual descriptor: lower case conversion, digit and punctuation removal, and English stop word removal. For *BoW* and *2grams*, we also apply Porter stemming.

The *word2vecSum* descriptor produces, for any input document, a vector corresponding to the sum of the word embedding vectors [93] related to each term within that document, while *word2vecAvg* computes the mean vector of them.

The doc2vec model promotes document-level embeddings for texts, and it is based on word embeddings [93], a preliminary work that assigns vector representations for words in order to capture their semantic relationships. *doc2vecApnews* stands for a 300-dimensional *doc2vec* [80] model, pre-trained over the Associated Press News textual dataset, of about 25M news articles from 2009 to 2015.

From the descriptors, rankers are defined as tuples of (descriptor, comparator), where the comparator corresponds to a dissimilarity function. We compose a ranker for each descriptor by choosing an appropriate comparator. Given that our method works on

Table 5.1: Datasets and descriptors for the experimental evaluation.

Dataset	Data	Descriptors	Evaluation Criteria
ME17-DIRSM	pictures, textual metadata	ResNet50IN, VGG16P365, NASNetIN, BoW, 2grams, doc2vecWiki	AP@[50,100,250,480] and mAP, for visual, textual, and multimodal scenarios
Brodatz	texture images	JCD, FCTH, CCOM	balanced accuracy, in a 80/20 split
Soccer	pictures	BIC, GCH, ACC	balanced accuracy, in a 80/20 split
UW	pictures, textual keywords	JAC, ACC, JCD, CEDD, word2vecSum, word2vecAvg, doc2vecWiki, doc2vecApnews	balanced accuracy, in a 80/20 split

top of rankers, we have to define dissimilarity functions to be used along with those descriptors that is not explicitly associated with one. This is the case for the four textual descriptors adopted in UW, as well as the descriptors adopted in ME17-DIRSM. All remaining descriptors define their own comparators.

For the textual descriptors BoW and 2grams, we adopt the Weighted Jaccard distance, defined as $1 - J(\mathbf{u}, \mathbf{v})$, where J is the Ruzicka similarity metric (Equation 5.1). Jaccard is a well-known and widely-used comparison metric for classic textual descriptors, specially for short texts, as in our case. For the remaining descriptors, we choose the Pearson correlation distance, defined as $1 - \rho(\mathbf{u}, \mathbf{v})$ (Equation 5.2), which is a general-purpose metric due to its suitability for highly dimensional data and scale invariance.

$$J(\mathbf{u}, \mathbf{v}) = \frac{\sum_i \min(u_i, v_i)}{\sum_i \max(u_i, v_i)} \tag{5.1}$$

$$\rho(\mathbf{u}, \mathbf{v}) = \frac{(\mathbf{u} - \bar{u}) \cdot (\mathbf{v} - \bar{v})}{\|(\mathbf{u} - \bar{u})\|_2 \|(\mathbf{v} - \bar{v})\|_2} \tag{5.2}$$

The datasets, descriptors, and evaluation criteria, are summarized in Table 5.1.

5.3.4 Fusion Setups

For both visual and textual scenarios in ME17-DIRSM, we analyze three variants of our method with respect to the input rankers for late-fusion. For the visual scenario, the combinations are ResNet50IN + NASNetIN, ResNet50IN + VGG16P365, and ResNet50IN + NASNetIN + VGG16P365. For the textual scenario, the combinations are BoW + 2grams, BoW + doc2vecWiki, and BoW + 2grams + doc2vecWiki.

As for the multimodal scenario, we investigate some combinations taking one ranker of each type, two of each, and three of each. Six multimodal combinations are evaluated: ResNet50IN + BoW, ResNet50IN + NASNetIN + BoW + 2grams, ResNet50IN + NASNetIN + BoW + doc2vecWiki, ResNet50IN + VGG16P365 + BoW + 2grams, ResNet50IN + VGG16P365 + BoW + doc2vecWiki, and ResNet50IN + NASNetIN + VGG16P365 + BoW + 2grams + doc2vecWiki.

We report three results for the adoption of *FV*, in its different embedding approaches,

Table 5.2: Base results of the chosen descriptors, along with a SVR regressor, in ME17-DIRSM.

(a) Visual.

Descriptor	AP@50	AP@100	AP@250	AP@480	mAP
ResNet50IN	100.00	98.90	98.02	85.92	95.71
NASNetIN	100.00	100.00	96.01	85.60	95.40
VGG16P365	100.00	97.74	93.65	84.59	94.00

(b) Textual.

Descriptor	AP@50	AP@100	AP@250	AP@480	mAP
BoW	81.85	78.62	72.29	65.51	74.57
2grams	82.01	76.58	73.63	65.40	74.43
doc2vecWiki	77.06	77.40	71.86	64.72	72.76

as a representation model for prediction tasks, in Soccer, Brodatz, and UW. We report the results for multiple descriptor combinations, in order to analyze: (i) the method against baselines, (ii) the embedding approaches, and (iii) the comparative effectiveness between the descriptor combinations. The descriptor combinations selected, although not exhaustive, are targeted for a large number of scenarios. In Soccer and Brodatz, all possible combinations were selected for evaluation. In UW, several visual combinations and multimodal combinations were selected. The descriptor combinations are:

- In Soccer: ACC + BIC, BIC + GCH, ACC + GCH, and ACC + BIC + GCH.
- In Brodatz: CCOM + FCTH, CCOM + JCD, FCTH + JCD, and CCOM + FCTH + JCD.
- In UW: ACC + CEDD, ACC + JCD, CEDD + JAC, CEDD + JCD, ACC + CEDD + JCD, and CEDD + JAC + JCD, for visual fusion, and ACC + doc2vecApnews, JCD + doc2vecApnews, ACC + JCD + doc2vecApnews, ACC + JCD + doc2vecWiki, ACC + JCD + word2vecAvg, and ACC + JCD+word2vecSum, for multimodal fusion.

5.3.5 Results and Discussion

Base Results

Here we report results by the use of individual descriptors. They constitute an initial baseline for our method as well as for other fusion approaches.

We report, in Tables 5.2a and 5.2b, the results for the visual and textual scenarios in ME17-DIRSM, achieved by the three visual and textual selected descriptors, along with a SVR regressor. As the task only mentioned AP@480 and mAP in their leaderboard, we focus our discussions on these two metrics. The correctness for the visual scenario is already high within these baselines, around 85% in AP@480. In the textual scenario, AP@480 is around 65%, which suggests more room for improvement.

We report, in Tables 5.3a, 5.3b, 5.3c, and 5.3d, the results obtained in Soccer, Brodatz, UW (visual), and UW (textual), respectively, by the use of the descriptors along with a SVM classifier.

Parameter Analysis

The resulting size of $\mathcal{G}$ is affected by the input rank sizes, defined by the hyper-parameter L. For the same reason, larger FG's either increase the vocabulary sizes of FV or the

Table 5.3: Base results obtained by the adoption the descriptors, along with a SVM classifier, in Soccer, Brodatz, and UW.

(a) Soccer

Descriptor	Balanced acc.
BIC	60.84
GCH	56.38
ACC	51.79

(b) Brodatz

Descriptor	Balanced acc.
JCD	78.65
FCTH	75.10
CCOM	74.65

(c) UW, visual

Descriptor	Balanced acc.
JAC	83.74
ACC	77.06
JCD	73.99
CEDD	72.46

(d) UW, textual

Descriptor	Balanced acc.
word2vecSum	86.91
word2vecAvg	82.96
doc2vecEnwi	82.95
doc2vecApnews	82.94

complexity to generate them. In Section 3.3.4, we showed that an increase in L leads to more discriminate graphs up to a saturation point. A practical upper bound for the choice of L tends to be the maximum rank size of users' interest, indirectly expressed here by the evaluation metrics.

As ME17-DIRSM defines evaluation metrics for ranks up to 480, we start by empirically evaluating the influence of L in the mAP score, for values to up 480. Figure 5.2 reports the influence of L for some of the elected fusion scenarios. The results were as expected: the effectiveness usually increased as L was larger. For the next evaluation scenarios in ME17-DIRSM, we adopt $L = 480$. For the other datasets, we adopt $L = 10$.

Fusion Results in Event Detection

We present our results achieved for the three scenarios in ME17-DIRSM, using the combinations proposed, along with the results of the 11 teams that participated in the competition. We also show the results achieved in [137] in the visual and multimodal scenarios, which relied on early-fusion techniques. These results are presented in Tables 5.4, 5.5, and 5.6, respectively for the visual, textual, and multimodal scenarios. In ME17-DIRSM, we focused on the $\mathcal{E}_V$ embedding approach. For the other datasets, we evaluate all of them.

In the visual scenario, only [2, 15] performed better, in terms of AP@480 and mAP, than our preliminary base setup, based on individual descriptors along with the SVR regressor. As for the textual scenario, only [125] in 12 initiatives surpassed BoW + SVR in AP@480, and [15, 99, 149] in mAP. This indicates that descriptors properly selected to the target problem can overcome more complex models, also requiring less effort.

Our method was superior in the visual scenario to all baselines, for two of three proposed variants of ranker combinations. Compared to the strongest baselines considering this scenario, our method presents gains from around 1 to 2% in AP@480, and 1% in mAP. Compared to the visual base results, from the individual descriptors, 3 to 4% in AP@480, and 1 to 2% in mAP.

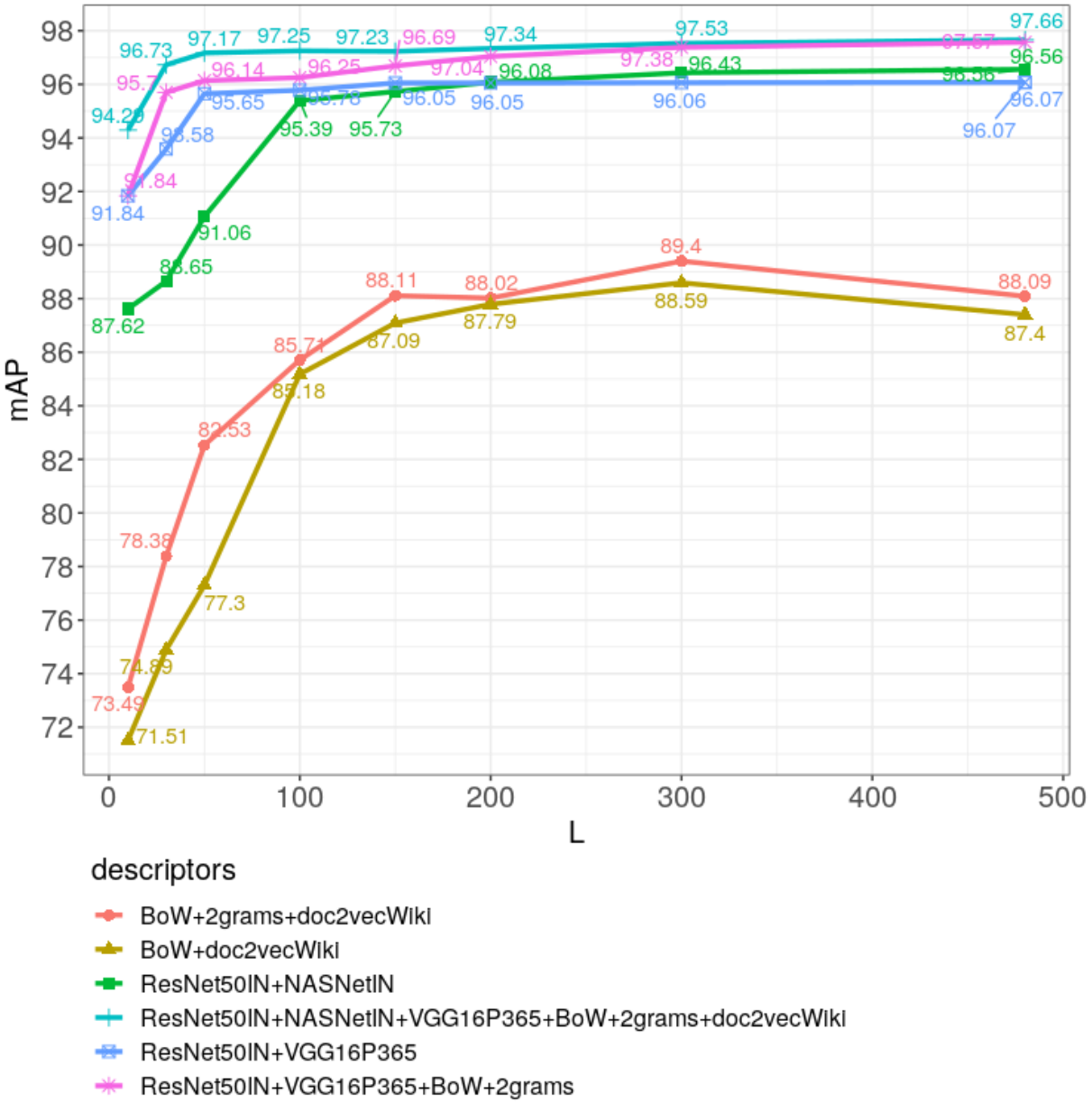


Figure 5.2: Effect of the rank size limit (L) for the fusion graph extraction, in the mAP score, for different fusion scenarios in ME17-DIRSM.

Table 5.4: Flood detection based on visual features, in ME17-DIRSM.

Method	AP@50	AP@100	AP@250	AP@480	mAP
FV-ResNet50IN+NASNetIN +VGG16P365	100.00	100.00	98.55	**88.41**	**96.74**
FV-ResNet50IN+NASNetIN	100.00	100.00	99.00	87.24	96.56
Ahmad et al. [2]				86.81	95.73
Bischke et al. [15]				86.64	95.71
FV-ResNet50IN+VGG16P365	100.00	100.00	97.89	86.40	96.07
Ahmad et al. [1]				84.94	95.11
Avgerinakis et al. [7]				78.82	92.27
Dao et al. [37]				77.62	87.87
Nogueira et al. [99]	96.20	93.69	87.30	74.67	87.96
Lopez-Fuentes et al. [86]				67.54	70.16
Hanif et al. [60]				64.90	80.98
Zhao and Larson [149]				51.46	64.70
Tkachenko et al. [125]				50.95	62.75
BoKG [137]	81.11				
BoCG [137]	47.94				
Fu et al. [54]					19.21

Table 5.5: Flood detection based on textual features, in ME17-DIRSM.

Method	AP@50	AP@100	AP@250	AP@480	mAP
FV-BoW+2grams +doc2vecWiki	100.00	93.88	84.67	**73.81**	**88.09**
FV-BoW+doc2vecWiki	97.56	93.16	83.47	73.74	86.98
FV-BoW+2grams	92.63	88.19	82.11	71.20	83.54
Tkachenko et al. [125]				66.78	74.37
Hanif et al. [60]				65.00	71.79
Zhao and Larson [149]				63.70	75.74
Bischke et al. [15]				63.41	77.64
Nogueira et al. [99]	88.24	84.41	72.61	62.80	77.02
Lopez-Fuentes et al. [86]				61.58	66.38
Dao et al. [37]				57.07	57.12
Avgerinakis et al. [7]				36.15	39.90
Ahmad et al. [1]				25.88	31.45
Ahmad et al. [2]				22.83	18.23
Fu et al. [54]					12.84

Table 5.6: Flood detection based on multimodal features, in ME17-DIRSM.

Method	AP@50	AP@100	AP@250	AP@480	mAP
FV-ResNet50IN+VGG16P365+BoW +doc2vecWiki	100.00	100.00	99.57	**90.96**	**97.63**
FV-ResNet50IN+NASNetIN +VGG16P365+BoW+2grams+doc2vecWiki	100.00	100.00	99.50	90.94	97.61
FV-ResNet50IN+VGG16P365+BoW +2grams	100.00	100.00	99.60	90.68	97.57
FV-ResNet50IN+NASNetIN+BoW +2grams	100.00	100.00	99.13	90.54	97.42
Bischke et al. [15]				90.45	97.40
FV-ResNet50IN+NASNetIN+BoW +doc2vecWiki	100.00	100.00	99.08	90.00	97.27
FV-ResNet50IN+BoW	100.00	100.00	99.11	89.09	97.05
Nogueira et al. [99]	100.00	100.00	97.76	85.85	95.90
Dao et al. [37]				85.41	90.39
Ahmad et al. [2]				83.73	92.55
Lopez-Fuentes et al. [86]				81.60	83.96
Zhao and Larson [149]				73.16	85.43
Tkachenko et al. [125]				72.26	80.87
Avgerinakis et al. [7]				68.57	83.37
Hanif et al. [60]				64.60	80.84
Ahmad et al. [1]				54.74	68.12
BoKG [137]	86.90				
BoCG [137]	73.85				
Fu et al. [54]					18.30

Table 5.7: Balanced accuracies by fusion methods in Soccer.

Method	ACC+BIC	BIC+GCH	ACC+GCH	ACC+BIC+GCH
FV-K	67.22	**67.09**	66.33	**68.88**
FV-V	**67.48**	64.29	**67.86**	68.62
FV-H	66.58	66.07	64.29	66.71
concatenation	62.50	62.76	55.48	63.39
majorityVote	—	—	—	62.76

Table 5.8: Balanced accuracies by fusion methods in Brodatz.

Method	CCOM+FCTH	CCOM+JCD	FCTH+JCD	CCOM+FCTH+JCD
FV-H	**89.59**	**91.84**	**88.13**	**91.65**
FV-V	88.56	90.98	87.29	91.10
FV-K	88.37	89.15	85.69	89.07
concatenation	88.16	89.18	86.73	88.76
majorityVote	—	—	—	83.42

Table 5.9: Balanced accuracies for visual fusion in UW.

Method	ACC + CEDD	ACC + JCD	CEDD + JAC	CEDD + JCD	ACC + CEDD + JCD	CEDD + JAC + JCD
FV-K	**85.88**	**85.02**	**84.83**	73.97	**83.97**	**85.73**
FV-H	84.58	84.95	84.47	74.13	80.50	81.66
FV-V	83.57	84.26	82.42	**74.16**	83.71	81.87
concatenation	82.75	82.64	77.70	70.28	83.18	82.23
majorityVote	—	—	—	—	76.22	73.05

Table 5.10: Balanced accuracies for multimodal fusion in UW.

Method	ACC + doc2vecApnews	JCD + doc2vecApnews	ACC + JCD + doc2vecApnews	ACC + JCD + doc2vecWiki	ACC + JCD + word2vecAvg	ACC + JCD + word2vecSum
FV-V	86.72	**87.21**	90.04	89.87	**92.84**	**92.84**
FV-K	88.52	85.52	**90.64**	**92.18**	92.51	92.65
FV-H	**90.26**	85.44	89.56	90.86	91.73	91.73
concatenation	88.69	86.57	90.38	88.28	90.73	90.87
majorityVote	—	—	83.77	83.11	84.62	84.49

In the textual scenario, our gains were even more expressive. It was superior in the textual scenario to all related works, for all three proposed variants of ranker combinations. Compared to the strongest baselines, our method presents gains from 5 to 7% in AP@480, and 6 to 11% in mAP. Compared to the textual base results, from the individual descriptors, 6 to 8% in AP@480, and 14 to 16% in mAP. In the multimodal scenario, considered baselines were more competitive. Again, however, our method presents gains over them, of 0.5% in AP@480 and 0.23% in mAP.

Fusion Results in Classification Tasks

We report in Tables 5.7, 5.8, 5.9, and 5.10, the results obtained by our method variants, respectively in Soccer, Brodatz, UW for image data, and UW for multimodal data, besides the results obtained by the baselines.

Our method led to significant gains when compared to the best base result from the descriptors in each dataset: around 8 p.p. in Soccer, 13.2 p.p. in Brodatz, 2.2 p.p. in UW for visual fusion, and 5.9 p.p. in UW for multimodal fusion. The gains in UW were comparatively lower than others, yet consistent, because the base results were already

higher, so that there were less room for improvement.

Our method, when compared to the best baseline in each descriptor combination in each dataset, had gains in all cases: up to 12.4 p.p. in Soccer, up to 2.9 p.p. in Brodatz, up to 7.1 p.p. in UW for visual fusion, and up to 3.9 p.p. in UW for multimodal fusion. Overall, all the FV approaches performed better than all baselines in all datasets. In only 6 of 20 descriptor combinations evaluated, any of the baselines surpassed any of the FV approaches.

The accuracy disparities, obtained by FV or any other fusion method, across the multiple descriptor combinations in each dataset, show that there is no obvious choice when dealing with which descriptors to be used together. As our representation model is meant to be unsupervised, this choice could only be guided by general heuristics, such as a selection of effective and low correlated descriptors, as discussed in Section 3.3.2. We leave this exploration for future work.

FV-K usually performed better than FV-H and FV-V, and FV-H usually performed better than FV-V, although in both cases the gains were at most 3 p.p one over the other. On the opposite side, FV-V is the simplest among the three, and FV-K requires more computational steps. These two aspects combined impose that the practical choice among the three must take into account the trade-off between accuracy vs computational cost. In any case, our method is unsupervised and feasible for general applicability.

5.4 Conclusions

In this chapter, we presented an unsupervised graph-based rank-fusion approach as a representation model for multimodal prediction tasks. Our solution is based on encoding multiple ranks into a graph representation, which is later embedded into a vectorial representation. Next, an estimator is built to predict if an input multimodal object refers to a target event or not, given their graph-based fusion representations.

The proposed method extends the fusion graphs – first introduced in Chapter 3 – for supervised learning tasks. It also applies a graph embedding mechanism in order to obtain the fusions vectors, a late-fusion vector representation that encodes multiple ranks and their inter-relationships automatically.

Performed experiments in multiple prediction tasks, such as flood detection and multimodal classification, demonstrate that our solution leads to highly effective results overcoming state-of-the-art solutions from both early-fusion and late-fusion underlying approaches.

Future work will focus on investigating the impact of semi-supervised and supervised approaches for the fusion graph and fusion vector constructions. We also plan to investigate the use of our solution in other multimodal problems, such as recommendation and hierarchical clustering. Finally, we plan to evaluate the proposed approach for other multimedia data, such as audio and video.

Chapter 6

Representation Learning for Fusion Vectors

6.1 Introduction

We have explored representation models based on ranks for general applicability, with successful applications in retrieval and prediction tasks. Most related works focused on solutions for a specific task, such as rank aggregation of data fusion. The concept of representation learning based on ranks is new in the literature.

In Section 6.2, we present learning approaches for optimized formulations for the fusion vectors – in contrast to its prior unsupervised definition from Chapter 4 – for the cases where labeled data are available. Next, we evaluate these approaches experimentally, in Section 6.3.

The main challenge concerning representation learning from ranks is to capture the semantics within ranks, in the presence of multiple heterogeneous rankers. The learning process should capture their complementarity while reducing redundancy. Besides, as the representation is the main goal, the learning function should be guided by constraints and factors that make that representation robust for general purpose use.

6.2 Proposed Framework

Let the embedding function $\mathcal{E} : \mathcal{G} \mapsto \mathcal{V}$ be a mapping, as previously defined in Section 2.1, where $\mathcal{G}$ is a fusion graph, $\mathcal{V}$ is a fusion vector, $\mathcal{V} \in \mathbb{R}^{\mathrm{d}}$, and d is the number of dimensions of a vector space (i.e., the vocabulary size). Also, let a rank-based representation model be the mapping $M : \mathcal{T} \mapsto \mathcal{V}$, such that, for any object q expressed by a rank set $\mathcal{T}$, we can obtain a fusion vector $\mathcal{V}$. If M is obtained through a learning process over a training set, then this process works as illustrated in Figure 6.1.

In Section 6.2.1, we introduce three learning approaches, which are based on feature engineering, embedding learning, and representation learning. In Section 6.2.2, we formalize the first learning approach, which is the focus of this chapter.

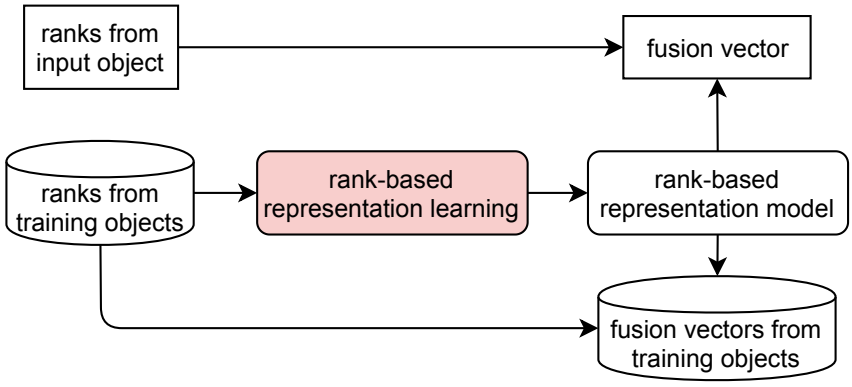


Figure 6.1: Conceptual learning process of a rank-based representation model, and its application to produce fusion vectors.

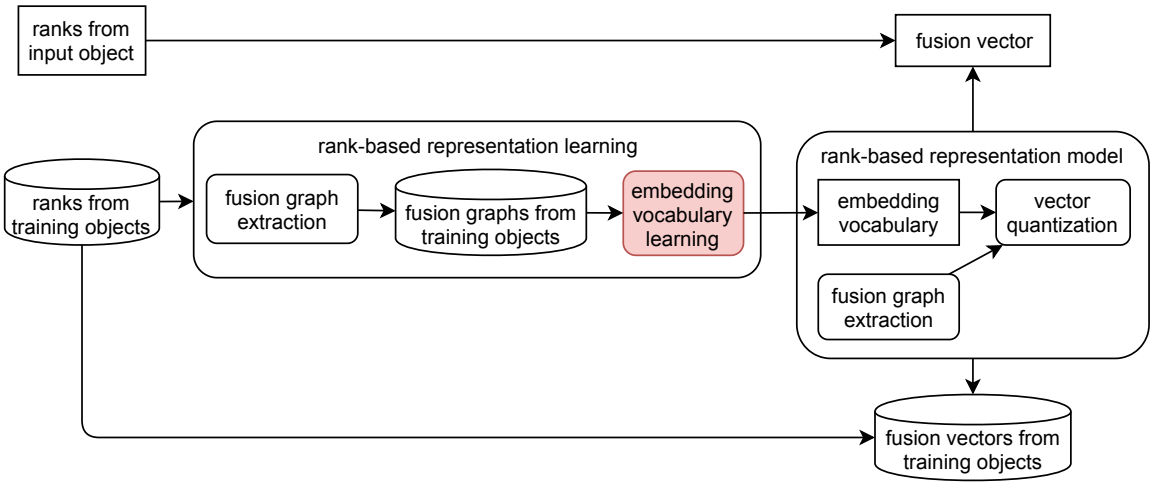


Figure 6.2: Inclusion of an embedding vocabulary learning step in a graph-based rank-based representation.

6.2.1 Alternatives for Implementation

The first approach concerns the introduction of feature engineering into the fusion vector generation procedure defined in Section 4.2. The objective is to optimize the feature selection during the vocabulary definition of the embedding function. We refer to this approach as ***embedding vocabulary learning***. Figure 6.2 illustrates the overall process involving the fusion vector generation, as in Chapter 4, and highlights the new proposed component, where *vector quantization* refers to assignment and pooling procedures. Despite this particular proposed module, the process is still mostly unsupervised.

A second approach refers to the development of a learning scheme, also based upon our fusion graphs, but for both the *embedding vocabulary learning* and *vector quantization* units from Figure 6.2. The advantage over the first approach is that the vector quantization unit could better fit the domain, instead of being defined a priori [63]. We refer to this approach as ***embedding learning***.

A third approach, referred to as ***rank-based representation learning***, aims at learning to embed objects by means of their ranks directly, i.e. it should learn an end-to-end rank-based representation model. That would correspond to the highlighted module in

Figure 6.1. This approach seems more straightforward, as the other two proposals still require the steps of fusion graph extraction and feature quantization. However, it poses a more challenging problem because it would require a customized learning procedure guided by a complex loss function. For this approach, we propose the investigation of end-to-end learning models, in which global composite loss functions could be defined to guide the whole optimization process [29, 59, 85]. That would comprehend internal units of feature extraction, vocabulary learning, and vector quantization, all of them learned at once.

6.2.2 Embedding Vocabulary Learning

Here we advance the first learning proposal, consisting of a feature selection approach for building the embedding vocabulary of rank-fusion graphs.

Our methods from the previous chapters focused so far on unsupervised learning. They adopted selection criteria based on frequent subgraphs, or direct graph projection based on vertex and edge statistics. While such approaches have been effective for graph embedding in multiple multimodal tasks, bringing prominent efficiency benefits without losing effectiveness, they have not yet exploited labeled data from datasets, not even optimization or reinforcement schemes.

Feature selection approaches for the graph domain is a demanding research field by itself. Most previous works regarding feature selection have focused on isolated features or feature sets. Graphs impose an additional challenge due to their complex nature.

We can achieve this goal by two possible strategies. The first one concerns the adoption of scoring functions for evaluating the feature candidates. It estimates the importance of the features, such that they can be sorted and reduced. Although this strategy uses labeled information, it does not involve a training paradigm. The second strategy involves post-processing and iterative techniques. The idea is to build an estimator for the training set, evaluate the feature contributions, and then reduce the feature set. This can be modeled as a search problem, such as the Recursive Feature Elimination (RFE) [58], or by simpler algorithms such as to compute the SHapley Additive exPlanations (SHAP) scores per feature and pruning the lower ones. That second strategy requires more computational cost, although it might lead to a better selection. Here we focus on the first of these strategies.

In order to define how to learn a vocabulary for embeddings, we propose an extension for the Bag of Graphs (BoG) [116], initially discussed in Section 4.2.2, in which its vocabulary (codebook) definition would now be guided by gains instead of either random selection or clustering. We refer to this proposal as Supervised Bag of Graphs (SBoG). By using labeled samples, we can exploit the concept of *feature importance* in order to assess the features' contributions to the vector space, and then propose a feature selection process. Iterative approaches can be exploited [51, 118, 143], in a way that they enable a proper selection of representative and discriminative features up to a certain convergence rate or desired number of iterations. IG, χ^2, and SHAP [87] are examples of possible feature importance metrics to be considered.

Supervised Kernel-based Embedding

Inspired by [8], which adapted χ^2 for term pairs in the context of text mining, here we define a χ^2 function for the graph domain, in the context of graph selection for embedding. Let GoI be a graph feature candidate, as we defined for BoG in Section 4.2.2. GoI is an undirected connected graph (V, E), whose vertices V are labeled, weighted in its vertices and edges, containing one central vertex v and a variable number of neighbor vertices n_i linked to v by paths of size 1. Let $K = \{n_1, \ldots, n_i \ldots, n_{|N|}\}$, $V = \{v\} \cup K$, $E = \{e_1, \ldots, e_j \ldots, e_{|N|}\}$, and $e_j = (v, n_i)$. In order to define χ^2 for GoI, first we extend it, from Equations 2.6 and 2.7, for any feature set s, as in Equations 6.1 and 6.2, where $\hat{A}$ is the number of samples, in c_i, which contain at least one element from s; $\hat{B}$ is the number of samples, in c_i, which do not contain any element from s; $\hat{C}$ is the number of samples, not in c_i, which contain at least one element from s; $\hat{D}$ is the number of samples, not in c_i, which do not contain any element from s; and $\hat{N} = \hat{A} + \hat{B} + \hat{C} + \hat{D}$. Given that, χ^2 is defined for GoI, by means of its vertices and edges, as in Equation 6.3.

$$\hat{\chi}^2(s, c_j) = \frac{\hat{N}(\hat{A}\hat{D} - \hat{B}\hat{C})^2}{(\hat{A} + \hat{C})(\hat{B} + \hat{D})(\hat{A} + \hat{B})(\hat{C} + \hat{D})}. \tag{6.1}$$

$$\hat{\chi}^2(s_i) = \max_{1 \leq j \leq |c|} (\hat{\chi}^2(s_i, c_j)). \tag{6.2}$$

$$\chi^2(GoI) = \begin{cases} \frac{\sum_{j=1}^{|E|} \hat{\chi}^2(\{v, e_j\})}{|E|} & \text{if } |E| > 0, \\ \chi^2(v) & \text{otherwise.} \end{cases} \tag{6.3}$$

Following the *relevance class frequency* intuition, explored in [133] for terms and term pairs, we extend this concept to the graph domain. Let $c(v)$ be the class frequency of the vertex v, i.e., the number of classes that present any graph sample containing the vertex v regardless its weight. Similarly, let $c(e)$ be the number of classes that present any graph sample containing the edge e regardless its weight. Given these definitions and Equation 2.8, we define $RCF(GoI)$, as expressed in Equation 6.4.

$$RCF(GoI) = \begin{cases} \frac{\min(c(v), \frac{\sum_{i=1}^{|K|} c(n_i)}{|K|})}{\frac{\sum_{j=1}^{|E|} c(e_j)}{|E|}} & \text{if } |E| > 0, \\ c(v) & \text{otherwise.} \end{cases} \tag{6.4}$$

In order to take into account both unsupervised and supervised feature selection strengths, we adopt a combined scoring function that employs both χ^2 and the relevance class frequency of a graph feature candidate. Hence, we define the final relevance score (RS) of a graph feature as expressed in Equation 6.5. The *log* factor is applied over RCF to penalize redundant features.

$$RS(GoI) = \chi^2(GoI) \times log(RCF(GoI)) \tag{6.5}$$

Based on this scoring function, we can adapt BoG in our preliminary FV-K embedding approach from Section 4.2.2. First, we extract the graph feature candidates from the graph

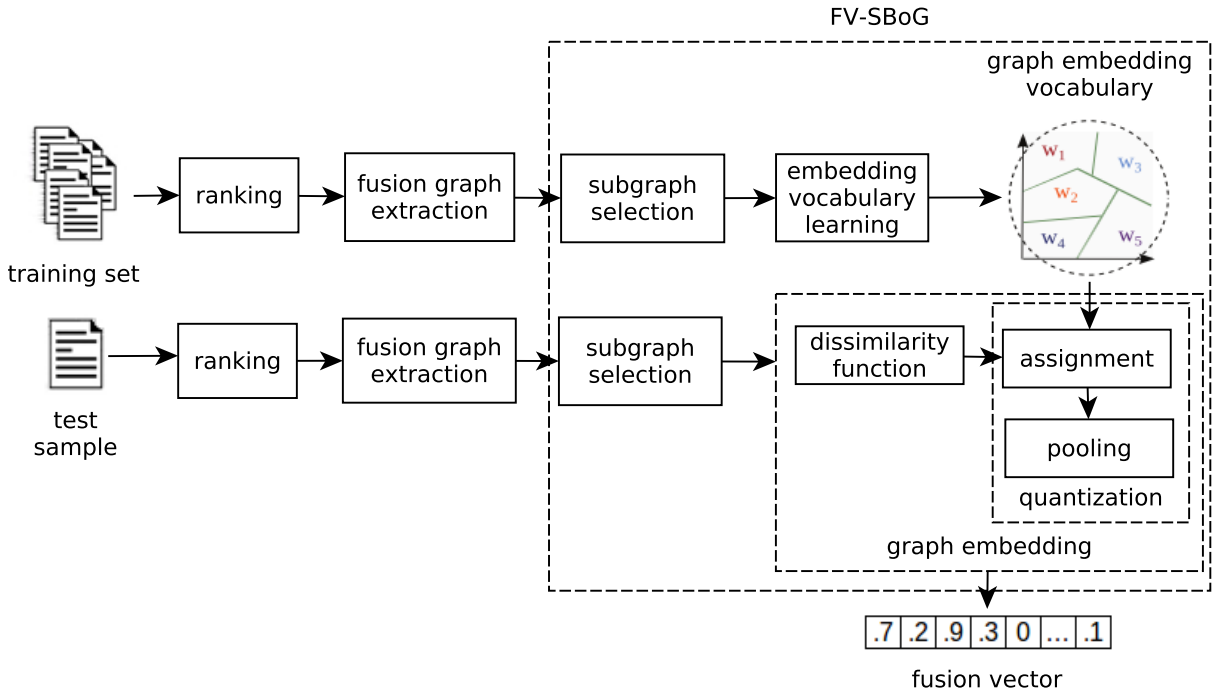


Figure 6.3: Supervised Bag of Graphs (SBoG) applied to vocabulary learning for the embedding of fusion graphs.

corpus. Then, we score them by Equation 6.5. The features with zero score are eliminated, and the remaining features are sorted in descending manner by their scores. From this point, we can retain either a fixed number of features to use, or a maximum proportion from the total (*maxProportion*). Other filtering criteria could be adopted, such as *Support* (*minSupp*), or a minimum threshold score per feature (*minScore*) [133]. These thresholds can be evaluated experimentally. The remaining of the BoG framework, as well as the embedding pipeline, remains the same as in Section 4.2.2 and Figure 6.2. We refer to this new embedding approach as FV-SBoG, summarized in Figure 6.3.

6.3 Experimental Evaluation

In order to evaluate FV-SBoG, we adopt the same evaluation procedure from Section 5.3.2, consisting of the evaluation of fusion approaches for multimodal classification tasks. Other scenarios could be explored, such as search tasks, as in Section 4.3.

The baselines, in this case, are mainly our previously defined unsupervised fusion approaches, which are based on the rank aggregation as fusion graphs followed by an embedding approach, for the representation model, plus a classifier model. Notice it that our own baselines here have already been superior to both state-of-the-art and classical early-fusion and late-fusion methods.

We elected the datasets Brodatz, Soccer, and UW, and two fusion setups for each:

- In Soccer: ACC + BIC, and ACC + GCH.
- In Brodatz: CCOM + FCTH, and FCTH + JCD.

- In UW: CEDD + JAC, and CEDD + JAC + JCD.

6.3.1 Parameter Analysis

In this section, we analyze the effects of *minSupp*, *minScore* and *maxProportion* in the classification performance achieved for each scenario. Although the optimum selection for these parameters may be dataset-dependant, they can be adjusted by a common grid search procedure over the training set. We also provide guidelines for their choice.

minSupp can be adopted to avoid the inclusion of too rare features, which are potentially non representative and noisy. We compute *minSupp* for a *GoI* graph feature candidate by means of its central vertex v. Therefore, for a certain *minSupp*, at least that number of samples from training corpus should contain v. We evaluate the values in $\{2, 4, 8\}$.

minScore can be applied as a threshold over the *RS(GoI)* scores, such that the features that don't achieve a score higher than the minimum are discarded. We evaluate the values in $\{0, 1, 2, 3, 4\}$.

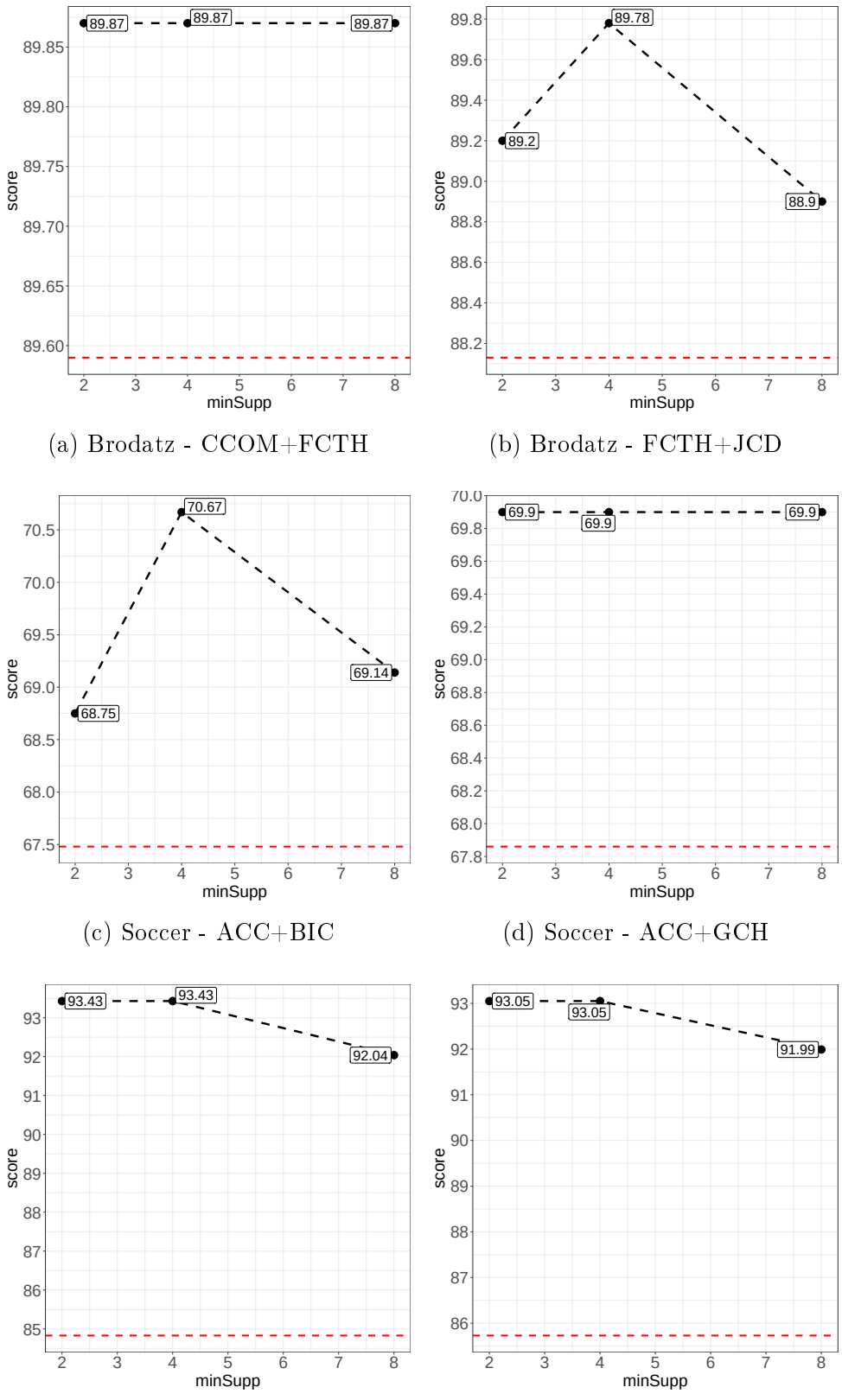
The *maxProportion* parameter establishes a feature selection rate. As the feature candidates are scored and ranked, if we keep taking features from the list we are subject to add valuable features up to a certain point. We evaluate the values, in percentages, in $\{40, 50, 60, 70, 80, 90, 100\}$.

Ideally, as long as the scoring function give us a good importance estimation, these parameters might be either avoided, merged, or even automatically adjusted. We plan to review these parameters, besides the scoring function and alternative formulations for it, in future work.

Figure 6.4 shows the influence of the *minSupp* parameter in the balanced accuracies achieved by FV-SBoG through the experimental scenarios. *minSupp* of 4 was the best, and values from 2 to 4 can be considered good choices. In some cases, its variation showed low or no influence. For datasets of larger number of samples per class, we could adopt a larger threshold.

Figure 6.5 shows the influence of the *minScore* parameter in FV-SBoG. Values of 2 and 3 are generally good choices. In practice, an optimum value for this threshold depends on how the scores assigned to the feature candidates are distributed. A score histogram analysis could be built in the training phase to possibly guide the parameter selection. Another selection for it is to apply grid search on possible values and select the best.

Despite the influence generated by the choices of *minSupp* and *minScore* in FV-SBoG, the results achieved are still much higher than the baselines. Among the model parameters, *maxProportion* is the most sensitive one. Figure 6.6 shows its influence in FV-SBoG. Best performances were achieved within $\{60, 70, 80, 90\}$, although the optimum choice depends on the dataset characteristics. We claim that an automatic yet straightforward approach can be adopted to help optimize the parameters. We plan to investigate this further in a future work.



(a) Brodatz - CCOM+FCTH

(b) Brodatz - FCTH+JCD

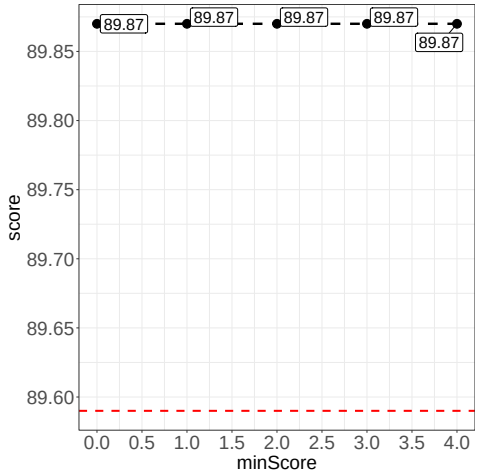
(c) Soccer - ACC+BIC

(d) Soccer - ACC+GCH

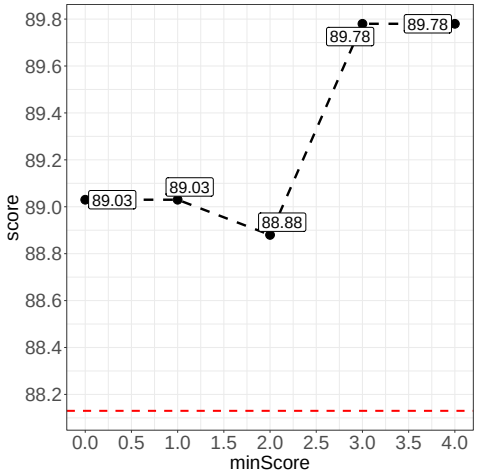
(e) UW - CEDD+JAC

(f) UW - CEDD+JAC+JCD

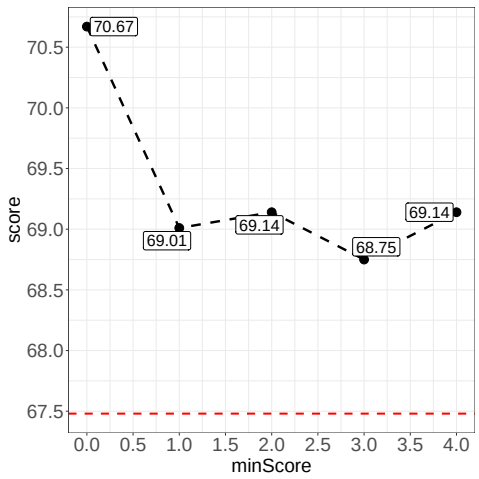
Figure 6.4: Effect of *minSupp* in the balanced accuracies obtained by FV-SBoG. The red lines denote the results from the strongest baselines per scenario.



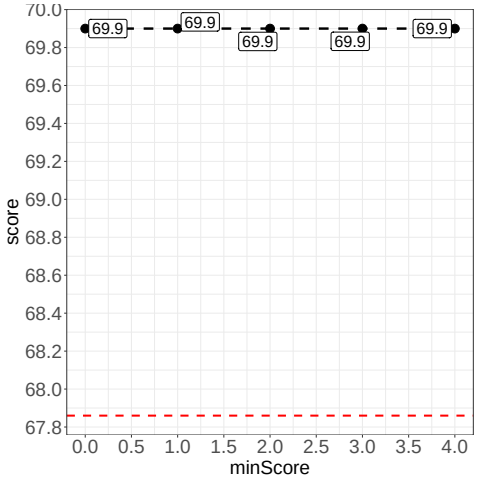
(a) Brodatz - CCOM+FCTH



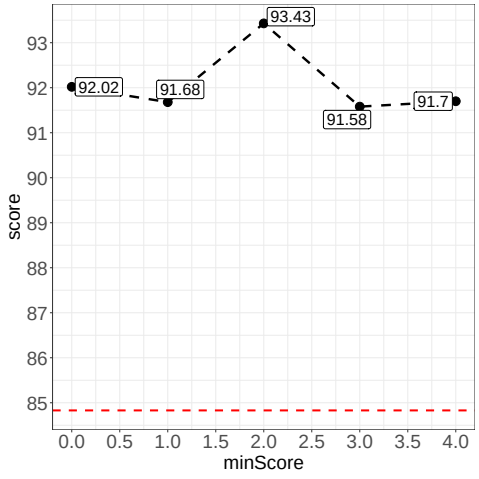
(b) Brodatz - FCTH+JCD



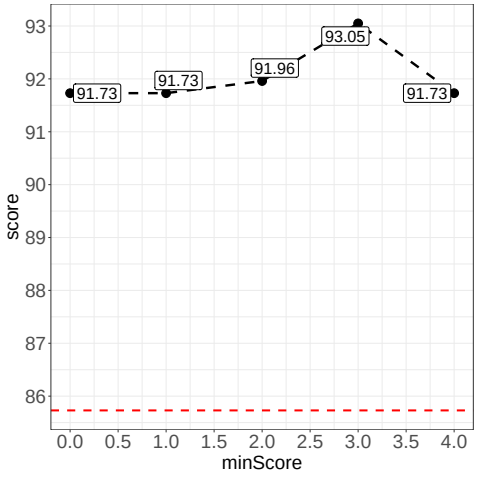
(c) Soccer - ACC+BIC



(d) Soccer - ACC+GCH

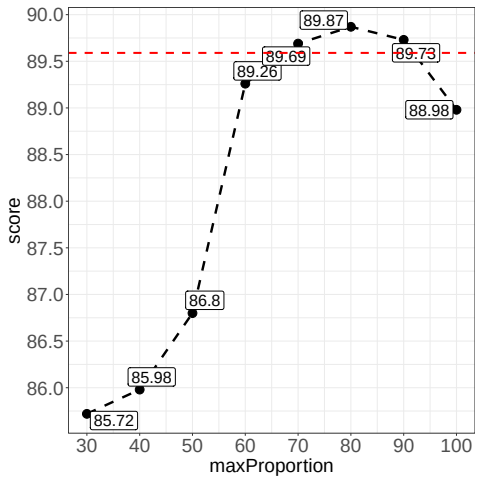


(e) UW - CEDD+JAC

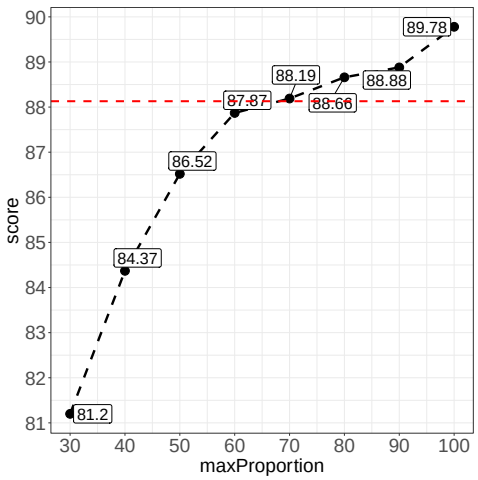


(f) UW - CEDD+JAC+JCD

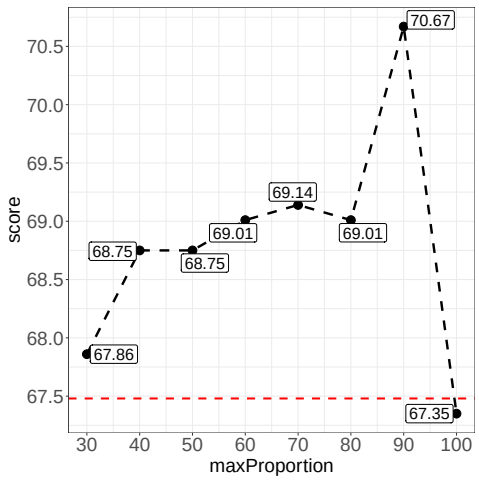
Figure 6.5: Effect of *minScore* in the balanced accuracies obtained by FV-SBoG. The red lines denote the results from the strongest baselines per scenario.



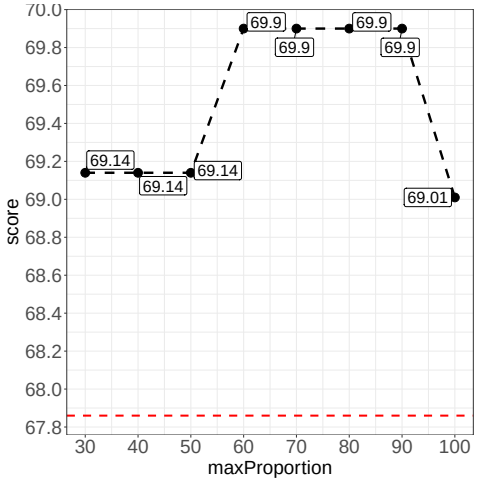
(a) Brodatz - CCOM+FCTH



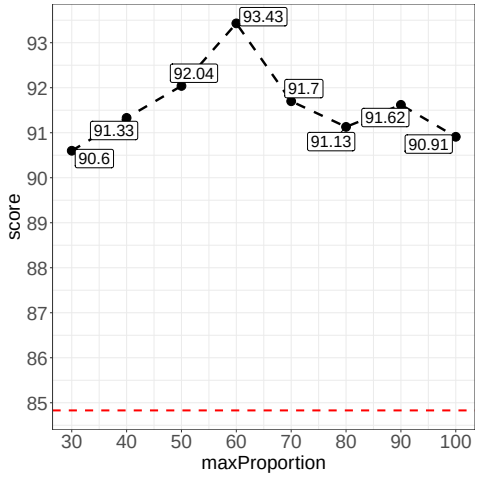
(b) Brodatz - FCTH+JCD



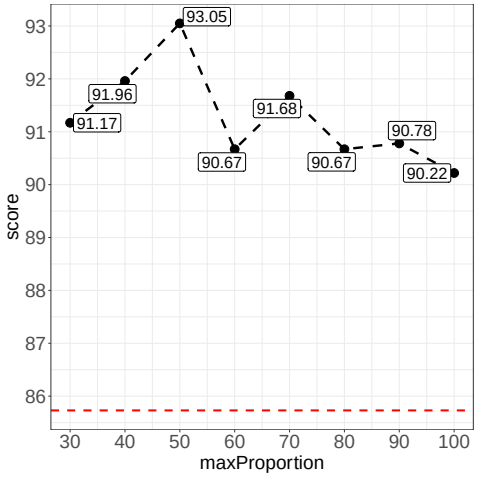
(c) Soccer - ACC+BIC



(d) Soccer - ACC+GCH



(e) UW - CEDD+JAC



(f) UW - CEDD+JAC+JCD

Figure 6.6: Effect of *maxProportion* in the balanced accuracies obtained by FV-SBoG. The red lines denote the results from the strongest baselines per scenario.

Table 6.1: Balanced accuracies by FV-SBoG and other fusion methods in Soccer.

Method	ACC+BIC	gains (in %)	ACC+GCH	gains (in %)
FV-SBoG	**70.67**	—	**69.90**	—
FV-K	67.22	5.13	66.33	5.38
FV-V	67.48	4.73	67.86	3.01
FV-H	66.58	6.14	64.29	8.73

Table 6.2: Balanced accuracies by FV-SBoG and other fusion methods in Brodatz.

Method	CCOM+FCTH	gains (in %)	FCTH+JCD	gains (in %)
FV-SBoG	**89.87**	—	**89.78**	—
FV-H	89.59	0.31	88.13	1.87
FV-V	88.56	1.48	87.29	2.85
FV-K	88.37	1.70	85.69	4.77

6.3.2 Fusion Results

Here we report the results achieved by FV-SBoG, in classification tasks, in contrast to its baselines. Tables 6.1, 6.2 and 6.3 report the results achieved respectively in Soccer, Brodatz, and UW. Gains of almost 14% were achieved over our unsupervised FV approaches, which are already strong when compared to fusion approaches from the literature.

This shows, once again, that representation models based on ranks are indeed a promising approach for fusion and multimodal tasks, even because this is still the first of our three learning frameworks proposed in Section 6.2.1. We plan to investigate and compare the other approaches in a future work.

6.4 Conclusions

In this work, we presented a variety of learning approaches that can be investigated in the context of rank-based representation models. Supervised learning and feature engineering have been massively explored in the literature, but their exploration in the graph domain, and specially in rank-fusion graph representations, is still a demanding research field.

We formally defined and evaluated one of those approaches, named FV-SBoG. Although it is still subject to improvement and deeper investigation, the results obtained so far show a novel and promising paradigm for fusion and multimodal tasks.

Concerning the work in this chapter, the main points we want to address in a future work are:

Table 6.3: Balanced accuracies by FV-SBoG and other fusion methods in UW.

	CEDD + JAC	gains (in %)	CEDD + JAC + JCD	gains (in %)
FV-SBoG	**93.43**	—	**93.05**	—
FV-K	84.83	10.14	85.73	8.54
FV-H	84.47	10.61	81.66	13.95
FV-V	82.42	13.36	81.87	13.66

- Alternative definitions for the scoring function of graphs;
- An automatic and straightforward strategy for parameter selection;
- The evaluation of FV-SBoG in search tasks;
- Formal definitions and experimental evaluation of the other learning approaches, named by us as *embedding learning* and *rank-based representation learning*;
- The proposal of extensions for semi-supervised scenarios.

Chapter 7

Conclusions

In this chapter, we summarize our main contributions. We also present possible extensions to be addressed in future work.

7.1 Main Contributions and Closing Remarks

The data landscape in large volume and heterogeneity brings numerous benefits to society, but also a number of technological challenges. There are effective solutions for handling this data in specific contexts, but we had missed approaches that would allow us to handle data from multiple sources in a unified manner independent of the target task. This motivated us to pursue multimodal models of general applicability.

There are, however, several practical challenges related to this goal. These data may be presented unlabeled or partially labeled, heterogeneous, unstructured, and derived from multiple modalities. One way we envisioned to unify how to deal with this data was to apply late fusion approaches, but only as part of the process, that is, we aimed to define a preliminary multimodal data integration approach to compose subsequent solutions. The way we planned to do so was to treat multiple descriptors, modalities, and retrieval models as components of a more robust approach. Thus, we decided to work with models based on query aggregation, always aiming to be able to represent objects for general purpose.

With the overall objective, and preliminary ideas in mind, we formalized the following hypothesis, so that it could guide the research and also be validated:

> ***Modeling objects using a graph-based representation, say a* fusion graph*, created based on information encoded on multiple ranks, leads to effective and efficient search and prediction systems.***

Given this hypothesis, we designed research questions that could help us investigate it through solid steps. We started with questions 1 and 2:

RQ$_1$ In a search scenario composed of multiple heterogeneous retrieval models at disposal, is it possible to define an unsupervised representation model, by means of ranks, to represent a query object as a graph, say *fusion graph*, that encodes its ranks and rank relationships effectively?

RQ$_2$ Is fusion graph an appropriate underlying structure to define a competitive unsupervised rank aggregation function when compared to state-of-the-art initiatives?

Regarding these two research questions, we made our first major contribution, focusing on the concept of *fusion graph*, which is a model capable of intrinsically encoding – for a digital object of any kind – information from it as well as comparatively to objects in a collection. We showed that graphs are suitable to encode ranks and their relationships. The model proved to be quite effective for content-based retrieval tasks, where we proposed a rank aggregation function that reshapes the ad-hoc retrieval problem as a fusion-graph retrieval problem. We showed that retrieval scores can be learned in unsupervised manner. This work was published as a journal article [45]:

- I. C. Dourado, D. C. G. Pedronette, and R. S. Torres. **Unsupervised graph-based rank aggregation for improved retrieval**. *Information Processing & Management*, 56(4):1260–1279, 2019.

Although we answered these first research questions, we had the initial impression that they could lead to answers that involved practical limitations. Therefore, we also elaborated the following complementary research question:

RQ$_3$ How to make graph-based rank aggregation functions efficient for search scenarios that require fast sub-linear retrieval times?

We addressed this research question in Chapter 4, where we proposed an efficient extension to the rank fusion retrieval framework. We concluded that vectors are able to represent objects by taking into account their ranks. This new model has maintained high levels of accuracy, incorporating expressive speedups, and boosting the solution for large-scale scenarios. From that chapter, our embedding and indexing proposals for rank-based graph representation also contribute for the rank aggregation and embedding literature. This work is under review in a journal. A preprint was made available [43]:

- I. C. Dourado and R. S. Torres. **Fusion Vectors: Embedding graph fusions for efficient unsupervised rank aggregation**. *arXiv:1906.06011*, 2019.

The answer to such questions has so far partially validated the hypothesis. The research questions related to prediction tasks, and the investigation of rank fusion models in labeled scenarios, as follows, were still missing:

RQ$_4$ Are graph-based rank representation models feasible for multimodal prediction tasks?

RQ$_5$ When labeled data is available, how could graph-based rank representation models be learned by training, by taking into account their discriminative power and efficiency?

The way we planned to work on these aspects was to investigate multimodal representation models. For this, we started from the previously defined fusion graphs and fusion

vectors, with the idea that they could be used to represent objects, whether labeled or not, in tasks such as prediction or clustering.

In Chapter 5, we addressed $\mathbf{RQ_4}$, in which we presented and evaluated a general purpose multimodal representation model that has proven effective and competitive. We concluded that ranks can be used to establish a general-purpose representation. Besides, graph-based rank-fusion models are promising for prediction tasks. The first results of this work were published in a conference paper [46]:

- I. C. Dourado, S. Tabbone, and R. S. Torres. **Event prediction based on unsupervised graph-based rank-fusion models**. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 88–98. Springer, 2019.

An extended version of it is under review in a journal. A preprint regarding this extension was made available [47]:

- I. C. Dourado, S. Tabbone, and R. S. Torres. **Multimodal representation model based on graph-based rank fusion**. *arXiv:1912.10314*, 2019.

In Chapter 6, we addressed $\mathbf{RQ_5}$, in which we presented possible different approaches to rank-based learning. We formalized and experimentally evaluated one of them, and we demonstrated that it is possible to further improve rank-based representations when a priori labeled information is available. Our investigation showed that feature engineering in graph embedding is a simple and effective approach to achieve concise, representative, and discriminative vectors.

In summary, graph-based rank fusion representations can be built from multiple heterogeneous rankers without any supervision. They allow the construction of robust adhoc retrieval models, and they also have general applicability for multimodal tasks. Finally, they can benefit from labeled data when provided.

Besides the publications about the core of the research, we also had the following publications during the development of the research:

- I. C. Dourado, R. Galante, M. A. Gonçalves, and R. S. Torres. Bag of Textual Graphs (BoTG): A general graph-based text representation model. *Journal of the Association for Information Science and Technology*, 70(8):817–829, 2019.

- R. O. Werneck, I. C. Dourado, S. G. Fadel, S. Tabbone, and R. S. Torres. Graph-Based Early-Fusion for Flood Detection. In *Proc. 25th IEEE International Conference on Image Processing*, pages 1048–1052. IEEE, 2018.

- K. Nogueira, S. G. Fadel, I. C. Dourado, R. O. Werneck, J. A. V. Muñoz, O. A. B. Penatti, R. T. Calumby, L. T. Li, J. A. dos Santos, and R. S. Torres. Exploiting ConvNet diversity for flooding identification. *IEEE Geoscience and Remote Sensing Letters*, 15(9):1446–1450, 2018.

- K. Nogueira, S. G. Fadel, I. C. Dourado, R. O. Werneck, J. A. V. Muñoz, O. A. B. Penatti, R. T. Calumby, L. T. Li, J. A dos Santos, and R. S. Torres. Data-Driven Flood Detection using Neural Networks. In *MediaEval Workshop*, Dublin, Ireland, 2017.

- R. T. Calumby, I. B. A. C. Araujo, F. S. Cordeiro, F. C. Bertoni, S. Canuto, F. Belém, M. A. Gonçalves, I. C. Dourado, J. A. V. Muñoz, L. T. Li, and R. S. Torres. Rank fusion and multimodal per-topic adaptiveness for diverse image retrieval. In *MediaEval Workshop*, Dublin, Ireland, 2017.

7.2 Future Work

Designing rank aggregation functions from rank fusion graphs or fusion vectors has proven to be a new and effective approach. Moreover, rank fusion representation models proved to be a novel and promising research field. Given this context, and even with the contributions obtained from this work, many research venues remain to be explored.

Besides possible advances and extensions presented at the end of each chapter, here we summarize the main future work possibilities, as well as ideas not yet previously enumerated:

1. *Extensions of the fusion graph generation for supervised and semi-supervised scenarios:* The fusion graph was introduced as a strategy to encode ranks and their relationships automatically, in unsupervised manner, in order to build rank aggregation functions, or to serve as a representation model. We claim that both the graph building, and the graph dissimilarity function, can be automatically adjusted or optimized for different situations, taking into consideration labeled data or ground truth relevance.

2. *Evaluation of our unsupervised rank aggregation functions against supervised methods:* Both FG and FV-based retrieval models, respectively from Chapter 3 and Chapter 4, have shown consistent effective retrieval results over the state of the art. A promising research venue would be to compare them with supervised methods from the literature, such as L2R techniques [96, 97], or semi-supervised rank aggregation functions [38, 109].

3. *Ablation study of the rank aggregation function based on fusion vectors:* In Chapter 4, a rank aggregation function for fast and accurate retrieval was presented. As it is composed of multiple components, one may evaluate alternative formulations for each one in the overall process, such as alternative fusion graph formulations [147], other embeddings approaches [154], and indexing paradigms [53].

4. *A joint rank-based representation model for both relevance and diversity:* Besides relevance, diversity is sometimes also considered a critical criterion for retrieval models, specially when the queries are related to multiple subjects [26, 26]. For this reason, possible future work would be to investigate and propose rank-based representation models capable of satisfying both objectives.

5. *Evaluation of the rank-based multimodal representation model in other scenarios:*. In Chapter 5, we discussed its application in event detection and multimodal classification. Given that multimodal scenarios are common and lack robust approaches,

another research venue refers to advancing this exploration for other services, such as recommendation or clustering.

6. *Extension of the fusion graphs and fusion vectors to other multimodal scenarios:* We claim such proposals can be extended for scenarios involving other types of data, especially temporal data such as audio and video. By doing this, we expand its applicability also for time series retrieval [50], cross-modal retrieval, and person re-identification [12], to name a few.

7. *The introduction of semi-supervised rank-based representation models:* In Chapter 6, we introduced FV-SBoG as a supervised proposal for building rank-based representation models when labeled data are present. The idea can also be explored for semi-supervised scenarios, in order to increase its applicability, while partially holding their discriminative power and reducing computational costs.

8. *The assessment FV-SBoG for retrieval and additional tasks:* While FV-SBoG is a supervised counterpart of a rank-based representation model originally introduced for search tasks, FV-SBoG itself was only evaluated for multimodal classification so far. Conversely, it can be applied for retrieval and many other scenarios involving multiple modalities or base retrieval models.

9. *The study of alternative scoring functions for FV-SBoG:* We presented a promising straightforward approach to assess graph feature importance. Many other formulations could be developed and evaluated as well. Feature importance for graph-based features is even an open problem by itself.

10. *The proposal of an automatic heuristic for FV-SBoG optimization:* The proposed approach relies on a few hyper-parameters. Another possible future work is to propose simple and efficient heuristics to guide their selection.

11. *Methods for rank-based representation learning:* In Chapter 6, we presented the notion of *rank-based representation learning*, and introduced possible approaches for it. As a future work, one may instantiate and validate a *rank-based representation learning*. It should consist of an end-to-end learning procedure to build a multimodal representation model that works directly on input ranks, intrinsically optimizing all internal parameters, including what modalities and rankers to adopt for the final representation.

12. *Methods for embedding learning: Embedding learning* for rank-based fusion graphs corresponds to another idea introduced in Chapter 6, but it still demands further development in terms of new methodologies for learning from graphs suitable vector representations. This problem is even more challenging if we consider time-evolving collections.

Bibliography

[1] K. Ahmad, K. Pogorelov, M. Riegler, N. Conci, and H. Pal. Cnn and gan based satellite and social media data fusion for disaster detection. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[2] S. Ahmad, K. Ahmad, N. Ahmad, and N. Conci. Convolutional neural networks for disaster images retrieval. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[3] J. A. Aledo, J. A. Gámez, and D. Molina. Using extension sets to aggregate partial rankings in a flexible setting. *Applied Mathematics and Computation*, 290(C):208–223, 2016.

[4] J. A. Aledo, J. A. Gámez, and A. Rosete. Approaching rank aggregation problems by using evolution strategies: The case of the optimal bucket order problem. *European Journal of Operational Research*, 270(3):982–998, 2018.

[5] S. Amodio, A. D'Ambrosio, and R. Siciliano. Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the kemeny axiomatic approach. *European Journal of Operational Research*, 249(2):667–676, 2016.

[6] N. Arica and F. T. Y. Vural. Bas: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9):1627–1639, 2003.

[7] K. Avgerinakis, A. Moumtzidou, S. Andreadis, E. Michail, I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris. Visual and textual analysis of social media and satellite images for flood detection@ multimedia satellite task mediaeval 2017. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[8] D. Badawi and H. Altınçay. A novel framework for termset selection and weighting in binary text classification. *Engineering Applications of Artificial Intelligence*, 35: 38–53, 2014.

[9] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley, Boston, MA, USA, 1999.

[10] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi. Feature selection using an improved chi-square for arabic text classification. *Journal of King Saud University-Computer and Information Sciences*, 2018.

[11] S. Bai and X. Bai. Sparse contextual activation for efficient visual re-ranking. *IEEE Transactions on Image Processing*, 25(3):1056–1069, 2016.

[12] S. Bai, X. Bai, Q. Tian, and L. J. Latecki. Regularized diffusion process on bidirectional context for object retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 41(5):1213–1226, 2018.

[13] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2018.

[14] A. Bhowmik and J. Ghosh. Letor methods for unsupervised rank aggregation. In *Proc. 26th World Wide Web Conference*, pages 1331–1340, Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[15] B. Bischke, P. Bhardwaj, A. Gautam, P. Helber, D. Borth, and A. Dengel. Detection of flooding events in social multimedia and satellite imagery using deep neural networks. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[16] B. Bischke, P. Helber, C. Schulze, S. Venkat, A. Dengel, and D. Borth. The multimedia satellite task at mediaeval 2017: Emergence response for flooding events. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[17] J. C. Borda. Mémoire sur les élections au scrutin. In *Histoire de l'Academie Royale des Sciences*, Paris, 1781.

[18] N. Bouhlel, G. Feki, A. B. Ammar, and C. B. Amar. Hypergraph learning with collaborative representation for image search reranking. *International Journal of Multimedia Information Retrieval*, pages 1–10, 2020.

[19] P. Brodatz. *Textures: A photographic album for artists and designers*. Dover, 1966.

[20] E. Bruni, N.-K. Tran, and M. Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

[21] H. Bunke and K. Riesen. Improving vector space embedding of graphs through feature selection algorithms. *Pattern Recognition*, 44(9):1928–1940, 2011.

[22] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3):255–259, 1998.

[23] H. Cai, V. W. Zheng, and K. C. C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[24] S. Canuto, D. X. Sousa, M. A. Gonçalves, and T. C. Rosa. A thorough evaluation of distance-based meta-features for automated text classification. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2242–2256, 2018.

[25] M. Carrillo, E. Villatoro-Tello, A. López-López, C. Eliasmith, M. Montes-y Gómez, and L. V. Pineda. Representing context information for document retrieval. In *Proc. 8th International Conference on Flexible Query Answering Systems*, pages 239–250, Berlin, Heidelberg, 2009. Springer.

[26] L. Chang, F. Haoyun, and d. R. Maarten. A contextual-bandit approach to online learning to rank for relevance and diversity. *arXiv:1912.00508*, 2019.

[27] S. A. Chatzichristofis and Y. S. Boutalis. Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In *International Conference on Computer Vision Systems*, pages 312–322. Springer, 2008.

[28] S. A. Chatzichristofis and Y. S. Boutalis. Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval. In *International Workshop on Image Analysis for Multimedia Interactive Services*, pages 191–196. IEEE, 2008.

[29] S.-Z. Chen, C.-C. Guo, and J.-H. Lai. Deep ranking for person re-identification via joint representation learning. *IEEE Transactions on Image Processing*, 25(5): 2353–2367, 2016.

[30] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[31] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5): 603–619, 2002.

[32] G. V. Cormack, C. L. A. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proc. 32nd ACM Special Interest Group on Information Retrieval*, pages 758–759. ACM, 2009.

[33] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision*, pages 1–22, 2004.

[34] L. Cui, Y. Jiao, L. Bai, L. Rossi, and E. R. Hancock. Adaptive feature selection based on the most informative graph-based features. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 276–287. Springer, 2017.

[35] A. D'Ambrosio, G. Mazzeo, C. Iorio, and R. Siciliano. A differential evolution algorithm for finding the median ranking under the kemeny axiomatic approach. *Computers and Operations Research*, 82(C):126–138, 2017.

[36] I.-S. Dao, P. Q. N. Minh, and A. Kasem. A context-aware late-fusion approach for disaster image retrieval from social media. In *Proc. 8th ACM International Conference on Multimedia Retrieval*, pages 266–273, New York, NY, USA, 2018. ACM.

[37] M. S. Dao, Q. N. M. Pham, D. Nguyen, and D. Tien. A domain-based late-fusion for disaster image retrieval from social media. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[38] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao. Visual reranking through weakly supervised multi-graph learning. In *Proc. IEEE International Conference on Computer Vision*, pages 2600–2607, 2013.

[39] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11(2):77–107, 2008.

[40] P. J. Dickinson, H. Bunke, A. Dadej, and M. Kraetzl. Matching graphs with unique node labels. *Pattern Analysis and Applications*, 7(3):243–254, 2004.

[41] X. Dong, Y. Yan, M. Tan, Y. Yang, and I. W. Tsang. Late fusion via subspace search with consistency preservation. *IEEE Transactions on Image Processing*, 28 (1):518–528, 2018.

[42] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1320–1327, 2013.

[43] I. C. Dourado and R. S. Torres. Fusion vectors: Embedding graph fusions for efficient unsupervised rank aggregation. *arXiv:1906.06011*, 2019.

[44] I. C. Dourado, R. Galante, M. A. Gonçalves, and R. S. Torres. Bag of textual graphs (botg): A general graph-based text representation model. *Journal of the Association for Information Science and Technology*, 70(8):817 – 829, 2019.

[45] I. C. Dourado, D. C. G. Pedronette, and R. S. Torres. Unsupervised graph-based rank aggregation for improved retrieval. *Information Processing & Management*, 56 (4):1260–1279, 2019.

[46] I. C. Dourado, S. Tabbone, and R. S. Torres. Event prediction based on unsupervised graph-based rank-fusion models. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 88–98. Springer, 2019.

[47] I. C. Dourado, S. Tabbone, and R. S. Torres. Multimodal representation model based on graph-based rank fusion. *arXiv:1912.10314*, 2019.

[48] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. 10th World Wide Web Conference*, pages 613–622, New York, NY, USA, 2001. ACM.

[49] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 301–312. ACM, 2003.

[50] F. A Faria, J. Almeida, B. Alberton, L. P. C. Morellato, and R. S. Torres. Fusion of time series representations for plant recognition in phenology studies. *Pattern Recognition Letters*, 83:205–214, 2016.

[51] F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. Meira Jr. Word co-occurrence features for text classification. *Information Systems*, 36(5): 843–858, 2011.

[52] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proc. 2nd Text REtrieval Conference*, pages 243–252, 1994.

[53] C. Fu, C. Xiang, C. Wang, and D. Cai. Fast approximate nearest neighbor search with the navigating spreading-out graph. *Proceedings of the VLDB Endowment*, 12 (5):461–474, 2019.

[54] X. Fu, Y. Bin, L. Peng, J. Zhou, Y. Yang, and H. T. Shen. Bmc@mediaeval 2017 multimedia satellite task via regression random forest. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[55] J. Gibert, E. Valveny, and H. Bunke. Graph embedding in vector spaces by node attribute statistics. *Pattern Recognition*, 45(9):3072–3083, 2012.

[56] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proc. 25th International Conference on Very Large Data Bases*, pages 518–529, 1999.

[57] R. Gopalan, P. Turaga, and R. Chellappa. Articulation-invariant representation of non-planar shapes. In *Proc. 11th European Conference on Computer Vision*, pages 286–299, Berlin, Heidelberg, 2010. Springer.

[58] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[59] K. Han, J. Guo, C. Zhang, and M. Zhu. Attribute-aware attention model for fine-grained representation learning. In *Proc. 26th ACM International Conference on Multimedia*, pages 2040–2048, New York, NY, USA, 2018. ACM.

[60] M. Hanif, M. A. Tahir, M. Khan, and M. Rafi. Flood detection using social media data and spectral regression based kernel discriminant analysis. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[61] B. S. Harish and M. B. Revanasiddappa. A comprehensive survey on various feature selection methods to categorize text documents. *International Journal of Computer Applications*, 164(8):1–7, 2017.

[62] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016.

[63] S. He, K. Liu, G. Ji, and J. Zhao. Learning to represent knowledge graphs with gaussian embedding. In *Proc. 24th ACM International Conference on Information and Knowledge Management*, pages 623–632, New York, NY, USA, 2015. ACM.

[64] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *Proc. ACM Special Interest Group on Information Retrieval*, pages 192–201. Springer, 1994.

[65] F. Hill and A. Korhonen. Learning abstract concept embeddings from multi-modal data: Since you probably can't see what i mean. In *Proc. Empirical Methods in Natural Language Processing*, pages 255–265, 2014.

[66] C.-B. Huang and Q. Liu. An orientation independent texture descriptor for image retrieval. In *Proc. International Conference on Communications, Circuits and Systems*, pages 772–776, 2007.

[67] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Computer Vision and Pattern Recognition*, pages 762–768. IEEE, 1997.

[68] G. Hubert, Y. Pitarch, K. Pinel-Sauvagnat, R. Tournier, and L. Laporte. Tournarank: When retrieval becomes document competition. *Information Processing & Management*, 54(2):252 – 272, 2018.

[69] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11, 2010.

[70] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. 22nd ACM International Conference on Multimedia*, pages 675–678, New York, NY, USA, 2014. ACM.

[71] P. Kaur, M. Singh, and G. S. Josan. Comparative analysis of rank aggregation techniques for metasearch using genetic algorithm. *Education and Information Technologies*, 22(3):965–983, 2017.

[72] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[73] D. Kiela and L. Bottou. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 36–45, 2014.

[74] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? *arXiv:1805.08974*, 2018.

[75] S. Kottur, R. Vedantam, J. MF Moura, and D. Parikh. Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *Computer Vision and Pattern Recognition*, pages 4985–4994, 2016.

[76] V. Kovalev and S. Volmer. Color co-occurrence descriptors for querying-by-example. In *Multimedia Modeling*, pages 32–38. IEEE, 1998.

[77] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[78] Z.-Z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann. Multimedia classification and event detection using double fusion. *Multimedia tools and applications*, 71(1): 333–347, 2014.

[79] L. J. Latecki, R. Lakamper, and T. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Computer Vision and Pattern Recognition*, volume 1, pages 424–429. IEEE, 2000.

[80] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[81] J. Lewis, S. Ossowski, J. Hicks, M. Errami, and H. R. Garner. Text similarity: an alternative way to search medline. *Bioinformatics*, 22(18):2298–304, 2006.

[82] S. Liang, I. Markov, Z. Ren, and M. de Rijke. Manifold learning for rank aggregation. In *Proc. 27th World Wide Web Conference*, pages 1735–1744, Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[83] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, 2007.

[84] H. Ling, X. Yang, and L. J. Latecki. Balancing deformability and discriminability for shape matching. In *Proc. 11th European Conference on Computer Vision*, pages 411–424. Springer, 2010.

[85] X. Liu, S. Zhang, T. Huang, and Q. Tian. E^2bows: An end-to-end bag-of-words model via deep convolutional neural network. *arXiv:1709.05903*, 2017.

[86] L. Lopez-Fuentes, J. van de Weijer, M. Bolanos, and H. Skinnemoen. Multi-modal deep learning approach for flood detection. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[87] S. M. Lundberg and S. I. Lee. A unified approach to interpreting model predictions. In *Proc. 31st International Conference on Neural Information Processing Systems*, pages 4765–4774, USA, 2017. Curran Associates Inc.

[88] L. F. G. Magalhães, M. A. Gonçalves, S. D. Canuto, D. H. Dalip, M. Cristo, and P. Calado. Quality assessment of collaboratively-created web content with no manual intervention based on soft multi-view generation. *Expert Systems with Applications*, 132:226–238, 2019.

[89] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[90] S. Maneewongvatana and D. M. Mount. It's okay to be skinny, if your friends are fat. In *Center for Geometric Computing 4th Annual Workshop on Computational Geometry*, volume 2, pages 1–8, 1999.

[91] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11 (6):703–715, 2001.

[92] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proc. 27th International Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

[93] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[94] T. Mitchell. *Machine Learning*. McGraw-Hill, New York, NY, USA, 1997.

[95] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proc. 11st ACM International Conference on Information and Knowledge Management*, pages 538–548, New York, NY, USA, 2002. ACM.

[96] A. Mourão and J. Magalhães. Low-complexity supervised rank fusion models. In *Proc. 27th ACM International Conference on Information and Knowledge Management*, pages 1691–1694, New York, NY, USA, 2018. ACM.

[97] J. A. V. Muñoz, R. S. Torres, and M. A. Gonçalves. A soft computing approach for learning to aggregate rankings. In *Proc. 24th ACM International Conference on Information and Knowledge Management*, pages 83–92, New York, NY, USA, 2015. ACM.

[98] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168. IEEE, 2006.

[99] K. Nogueira, S. G. Fadel, I. C. Dourado, R. O. Werneck, J. A. V. Muñoz, O. A. B. Penatti, R. T. Calumby, L. T. Li, J. A dos Santos, and R. S. Torres. Data-driven flood detection using neural networks. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[100] K. Nogueira, S. G. Fadel, I. C. Dourado, R. O. Werneck, J. A. V. Muñoz, O. A. B. Penatti, R. T. Calumby, L. T. Li, J. A. dos Santos, and R. S. Torres. Exploiting convnet diversity for flooding identification. *IEEE Geoscience and Remote Sensing Letters*, 15(9):1446–1450, 2018.

[101] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[102] G. Park and W. Im. Image-text multi-modal representation learning by adversarial backpropagation. *arXiv:1612.08354*, 2016.

[103] D. C. G. Pedronette and R. S. Torres. Shape retrieval using contour features and distance optimization. In *Proc. International Conference on Computer Vision Theory and Applications*, volume 1, pages 197–202, 2010.

[104] D. C. G. Pedronette and R. S. Torres. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, 46(8):2350–2360, 2013.

[105] D. C. G. Pedronette and R. S. Torres. A correlation graph approach for unsupervised manifold learning in image retrieval tasks. *Neurocomputing*, 208:66–79, 2016.

[106] D. C. G. Pedronette, J. Almeida, and R. S. Torres. A graph-based ranked-list model for unsupervised distance learning on shape retrieval. *Pattern Recognition Letters*, 83:357–367, 2016.

[107] D. C. G. Pedronette, F. M. F. Gonçalves, and I. R. Guilherme. Unsupervised manifold learning through reciprocal knn graph and connected components for image retrieval tasks. *Pattern Recognition*, 75:161–174, 2018.

[108] D. C. G. Pedronette, L. P. Valem, J. Almeida, and R. S. Torres. Multimedia retrieval through unsupervised hypergraph-based manifold ranking. *IEEE Transactions on Image Processing*, 28(12):5824–5838, 2019.

[109] D. C. G. Pedronette, Y. Weng, A. Baldassin, and C. Hou. Semi-supervised and active learning through manifold reciprocal knn graph for image retrieval. *Neurocomputing*, 340:19–31, 2019.

[110] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *Computer Vision and Pattern Recognition*, pages 777–784. IEEE, June 2011.

[111] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Text REtrieval Conference*, pages 109–126, 1994.

[112] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. B., A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[113] A. Schenker, H. Bunke, M. Last, and A. Kandel. Clustering of web documents using graph representations. In *Applied Graph Theory in Computer Vision and Pattern Recognition*, pages 247–265. Springer, 2007.

[114] D. Sculley. Rank aggregation for similar items. In *SIAM International Conference on Data Mining*, pages 587–592, 2007.

[115] C. Silberer and M. Lapata. Learning grounded meaning representations with autoencoders. In *Proc. 52nd Annual Meeting of the Association for Computational Linguistics*, pages 721–732, 2014.

[116] F. B. Silva, R. O. Werneck, S. Goldenstein, S. Tabbone, and R. S. Torres. Graph-based bag-of-words for classification. *Pattern Recognition*, 74:266 – 285, 2018.

[117] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[118] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, pages 73–86. Springer, 2012.

[119] F. Song, S. Liu, and J. Yang. A comparative study on text representation schemes in text categorization. *Pattern analysis and applications*, 8(1-2):199–209, 2005.

[120] D. X. Sousa, S. Canuto, M. A. Gonçalves, T. C. Rosa, and W. S. Martins. Risk-sensitive learning to rank with evolutionary multi-objective feature selection. *ACM Transactions on Information Systems*, 37(2):24, 2019.

[121] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *Proc. 11st ACM International Conference on Information and Knowledge Management*, pages 102–109. ACM, 2002.

[122] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[123] B. Tao and B. W. Dickinson. Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation*, 11(3):327–342, 2000.

[124] N. Tax, S. Bockting, and D. Hiemstra. A cross-benchmark comparison of 87 learning to rank methods. *Information Processing & Management*, 51(6):757 – 772, 2015.

[125] N. Tkachenko, A. Zubiaga, and R. Procter. Wisc at mmediaeval 2017: Multimedia satellite task. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[126] R. S. Torres and A. X. Falcao. Content-based image retrieval: theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.

[127] R. S. Torres and A. X. Falcão. Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing*, 25(1):3–13, 2007.

[128] C. Tzelepis, Z. Ma, V. Mezaris, B. Ionescu, I. Kompatsiaris, G. Boato, N. Sebe, and S. Yan. Event-based media processing and analysis: A survey of the literature. *Image and Vision Computing*, 53:3–19, 2016.

[129] L. P. Valem and D. C. G. Pedronette. Selection and combination of unsupervised learning methods for image retrieval. In *Proc. of the 15th International Workshop on Content-Based Multimedia Indexing*, page 27. ACM, 2017.

[130] J. Van De Weijer and C. Schmid. Coloring local feature extraction. *European Conference on Computer Vision*, pages 334–348, 2006.

[131] J. C. Van Gemert, C. J. Veenman, A. W. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, 2010.

[132] W. D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray. Graph distances using graph union. *Pattern Recognition Letters*, 22(6):701–704, 2001.

[133] C. Wan, Y. Wang, Y. Liu, J. Ji, and G. Feng. Composite feature extraction and selection for text classification. *IEEE Access*, 7:35208–35219, 2019.

[134] B. Wang, J. Jiang, W. Wang, Z. H. Zhou, and Z. Tu. Unsupervised metric fusion by cross diffusion. In *25th Computer Vision and Pattern Recognition*, pages 2997–3004. IEEE, June 2012.

[135] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han. Contextual weighting for vocabulary tree based image retrieval. In *Proc. IEEE International Conference on Computer Vision*, pages 209–216. IEEE, 2011.

[136] Y. Wang, L. Zhu, X. Qian, and J. Han. Joint hypergraph learning for tag-based image retrieval. *IEEE Transactions on Image Processing*, 27(9):4437–4451, 2018.

[137] R. O. Werneck, I. C. Dourado, S. G. Fadel, S. Tabbone, and R. S. Torres. Graph-based early-fusion for flood detection. In *Proc. 25th IEEE International Conference on Image Processing*, pages 1048–1052. IEEE, 2018.

[138] A. Williams and P. Yoon. Content-based image retrieval using joint correlograms. *Multimedia Tools and Applications*, 34(2):239–248, 2007.

[139] P. Wu, B. S. Manjunanth, S. D. Newsam, and H. D. Shin. A texture descriptor for image retrieval and browsing. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 3–7, 1999.

[140] L. Xie, R. Hong, B. Zhang, and Q. Tian. Image classification and retrieval are one. In *Proc. 5th ACM International Conference on Multimedia Retrieval*, pages 3–10, New York, NY, USA, 2015. ACM.

[141] Y. Xu, G. J. Jones, J. Li, B. Wang, and C. Sun. A study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems*, 3(3):1007–1012, 2007.

[142] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang. Network representation learning with rich text information. In *Proc. 24th International Joint Conference on Artificial Intelligence*, 2015.

[143] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Computer Vision and Pattern Recognition*, pages 4042–4049. IEEE, 2014.

[144] S. E. Yuksel, J. N. Wilson, and P. D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.

[145] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. Tensor fusion network for multimodal sentiment analysis. *arXiv:1707.07250*, 2017.

[146] K. Zagoris, S. A. Chatzichristofis, N. Papamarkos, and Y. S. Boutalis. Automatic image annotation and retrieval using the joint composite descriptor. In *2010 14th Panhellenic Conference on Informatics*, pages 143–147. IEEE, 2010.

[147] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific rank fusion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):803–815, April 2015.

[148] W. Zhao, J. Mao, and K. Lu. Ranking themes on co-word networks: Exploring the relationships among different metrics. *Information Processing & Management*, 54 (2):203 – 218, 2018.

[149] Z. Zhao and M. Larson. Retrieving social flooding images based on multimodal information. In *MediaEval Workshop*, Dublin, Ireland, 2017.

[150] L. Zheng, S. Wang, and Q. Tian. $\mathcal{L}_p$-norm idf for scalable image retrieval. *IEEE Transactions on Image Processing*, 23(8):3604–3617, Aug 2014.

[151] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian. Query-adaptive late fusion for image search and person re-identification. In *Proc. 28th Computer Vision and Pattern Recognition*, pages 1741–1750. IEEE, June 2015.

[152] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[153] W. Zhou, H. Li, and Q. Tian. Recent advance in content-based image retrieval: A literature survey. *arXiv:1706.06064*, 2017.

[154] Y. Zhu, J. X. Yu, and L. Qin. Leveraging graph dimensions in online graph search. *Proc. 40th International Conference on Very Large Data Bases*, 8(1):85–96, 2014.

[155] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *arXiv:1707.07012*, 2017.